

TUGAS AKHIR - KI141502

**RANCANG BANGUN APLIKASI *FACESHIFT*
MENGUNAKAN *MOTION CAPTURE* PADA
WAJAH DENGAN TEKNOLOGI *INTEL*
*REALSENSE***

**ISHARDAN
NRP 5113100182**

**Dosen Pembimbing
Dr.Eng Darlis Herumurti, S.Kom.,M.Kom.
Anny Yuniarti, S.Kom., M.Comp.Sc.**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017**



TUGAS AKHIR - KI141502

**RANCANG BANGUN APLIKASI *FACESHIFT*
MENGUNAKAN *MOTION CAPTURE* PADA
WAJAH DENGAN TEKNOLOGI *INTEL*
*REALSENSE***

**ISHARDAN
NRP 5113100182**

**Dosen Pembimbing
Dr.Eng Darlis Herumurti, S.Kom.,M.Kom.
Anny Yuniarti, S.Kom., M.Comp.Sc.**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017**

(Halaman ini sengaja dikosongkan)



FINAL PROJECT- KI141502

**RANCANG BANGUN APLIKASI *FACESHIFT*
MENGUNAKAN *MOTION CAPTURE* PADA
WAJAH DENGAN TEKNOLOGI *INTEL REALSENSE***

**ISHARDAN
NRP 5113100182**

**Dosen Pembimbing
Dr.Eng Darlis Herumurti, S.Kom.,M.Kom.
Anny Yuniarti, S.Kom., M.Comp.Sc.**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017**

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

RANCANG BANGUN APLIKASI *FACESHIFT* MENGUNAKAN *MOTION CAPTURE* PADA WAJAH DENGAN TEKNOLOGI INTEL *REALSENSE*

Tugas Akhir

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer pada
Rumpun Mata Kuliah Interaksi, Grafika, dan Seni
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

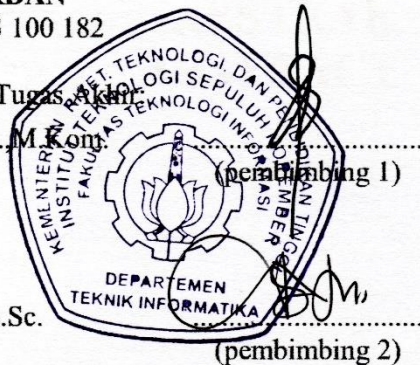
ISHARDAN

NRP. 5113 100 182

Disetujui oleh Dosen Pembimbing Tugas Akhir

Dr.Eng Darlis Herumurti, S.Kom., M.Kom.

NIP: 197712172003121001



(pembimbing 1)

Anny Yuniarti, S.Kom., M.Comp.Sc.

NIP: 198106222005012002

(pembimbing 2)

**SURABAYA
JUNI, 2017**

(Halaman ini sengaja dikosongkan)

RANCANG BANGUN APLIKASI *FACESHIFT* MENGUNAKAN *MOTION CAPTURE* PADA WAJAH DENGAN TEKNOLOGI INTEL *REALSENSE*

Nama Mahasiswa : Ishardan
NRP : 5113100182
Jurusan : Teknik Informatika FTIf-ITS
Dosen Pembimbing I : Dr.Eng Darlis Herumurti, S.Kom.,M.Kom.
Dosen Pembimbing II : Anny Yuniarti, S.Kom., M.Comp.Sc.

ABSTRAK

Perkembangan olah citra digital telah mengalami peningkatan yang sangat signifikan pada awal tahun 2010. Sebelumnya, citra digital khususnya animasi komputer pertama kali bersifat sederhana. Dengan majunya teknologi komputer, Citra digital telah dapat membuat film animasi dengan kualitas gambar yang mendekati nyata. Terdapat 2 cara membuat ekspresi wajah pada film animasi yaitu dengan manual dan menggunakan set kamera animator dengan biaya terbilang mahal.

Intel Realsense 3D Camera adalah teknologi kamera yang mampu merespon tangan, lengan, dan gerakan kepala serta ekspresi wajah. Kemampuan kamera Intel Realsense ini dapat mendeteksi kedalaman objek secara 3 Dimensi.

Pada Tugas akhir ini, Penulis membuat aplikasi faceshift dengan menggunakan teknologi Intel Realsense. Aplikasi ini mendeteksi wajah menambahkan poin landmark pada wajah pengguna kemudian data wajah disimpan dan kemudian digunakan untuk menggerakkan wajah karakter. Selain itu tugas akhir ini juga dapat mendeteksi dan menampilkan ekspresi dari pengguna melalui pergerakan emosi pada wajah.

Dari hasil ujicoba terhadap beberapa koresponden pada aplikasi ini, aplikasi yang dibangun berhasil memberikan pengalaman

baru kepada pengguna. Aplikasi ini dapat mendeteksi dan menggerakkan karakter sesuai dengan pergerakan pengguna. Selain itu aplikasi ini juga sudah dapat mendeteksi emosi dari pengguna walaupun akurasi masih harus ditingkatkan lagi pada hardware kamera intel realsense. Dengan dikembangkan aplikasi ini, diharapkan bisa memudahkan pengguna untuk membuat ekspresi wajah karakter yang lebih baik dan lebih natural.

Key words: Face Tracking, Intel Realsense, Musik, User Experience, Avatar.

DEVELOPMENT OF FACESHIFT APPLICATION USING INTEL REALSENSE CAMERA AND INTEL REALSENSE SDK

Student Name : Ishardan
NRP : 5113100182
Major : Informatics Engineer FTIf-ITS
Advisor I : Dr.Eng Darlis Herumurti, S.Kom.,M.Kom.

ABSTRACT

The development of image processing has been significant enhancement in early 2010. Previously, the first image processing was very simple. With the advent of information technology, the image processing has been able to create animated films with close-to-real picture quality.

When making animated films, it has specific difficulty than making an ordinary film, which is a natural face movement. There are 2 techniques to create a good facial expression, by manual or by using a set of animator cameras that cost quite expensive.

Intel Realsense 3D Camera is a camera technology that has capable to respond hands, arms, and head movements and facial expressions. Intel Realsense camera capabilities can detect the depth of objects in 3 Dimensions.

In this final project, the author tries to make faceshift application using Intel Realsense technology. This app works by adding the landmark points on the user's face then data stored and then used to move the character's face. In addition, this final project can also detect and display user expression by the movement of emotion on user's face.

From the results of trials of several correspondents in this application, built applications successfully provide new experiences to users. This app can detect and animates the characters according to user movement. And, this application is also able to detect the

emotions of the user although the accuracy still need to be improved again on intel realsense camera. With the developed application ii, is expected to facilitate the user to make the facial expression of a better and more natural character.

Key words: Face Tracking, Intel Realsense, Musik, User Experience, Avatar

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji dan syukur ke hadirat Allah Subhanahuwata'ala yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “Rancang Bangun Aplikasi *Faceshift* Menggunakan *Motion Capture* pada Wajah dengan Teknologi *Intel Realsense*”.

Pengerjaan tugas akhir ini adalah saat bagi penulis untuk mengeluarkan seluruh ilmu yang telah didapat semenjak di awal perkuliahan sampai hari ini di lingkungan jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya

Dalam pelaksanaan dan pembuatan tugas akhir ini, penulis menerima banyak sekali bantuan dari berbagai pihak, baik secara moril maupun materi. Melalui lembaran kata pengantar ini, penulis secara khusus menyampaikan ucapan terima kasih kepada:

1. Allah Subhanahu wa ta'ala yang telah melimpahkan kesehatan, semangat, dan hidayah-Nya sehingga penulis Alhamdulillah mampu menyelesaikan tugas akhir dengan baik.
2. Junjungan kita Nabi Muhammad SAW yang telah menjadi inspirasi, contoh yang baik bagi penulis sehingga tetap termotivasi dalam mengerjakan tugas akhir.
3. Kedua orang tua penulis yang telah mencurahkan doa, dukungan semangat, perhatian, serta kasih sayang kepada penulis.
4. Dr.Eng Darlis Herumurti, S.Kom.,M.Kom.. dan Ibu Anny Yuniarti, S.Kom., M.Comp.Sc. selaku dosen pembimbing yang senantiasa memberikan arahan.

5. Search engine Google yang melimpahkan penulis terhadap informasi-informasi penting terkait pembuatan aplikasi ini.
6. Didit Septiyano, yang sudah membantu penggunaan kamera *intel realsense* sehingga penulis dapat mengerjakan aplikasi ini.
7. Fajar Ade Putra, Naufal Bayu Fauzan, Abdurachman Rafif Sayudha, Guruh Arya Senna, Tikva Immanuel Mooy, Faizzudarain Syam, Fahmi Thor, Panji Rimawan, Shafly serta teman-teman sesama dalam perjuangan di laboratorium Interaksi Grafis dan Seni, telah memberikan arahan dan semangat dalam pengerjaan tugas akhir ini.
8. Teman-teman seangkatan yang selalu siap sedia ketika penulis mengalami kesulitan.
9. Teman-teman angkatan 2013, 2014 dan 2015 yang sudah memberikan pengalaman selama kuliah di Teknik Informatika ini.
10. Juga tidak lupa kepada semua pihak yang belum sempat disebutkan satu per satu yang telah membantu terselesaikannya tugas akhir ini, penulis mengucapkan terima kasih.

Penulis telah mengerjakan sebaik mungkin dalam penyusunan aplikasi tugas akhir ini. Penulis juga memohon maaf apabila terdapat kekurangan, kesalahan, maupun kelalaian yang telah penulis lakukan.

Surabaya, 4 Juni 2017

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	vii
ABSTRAK	ix
ABSTRACT	xi
DAFTAR ISI	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Pembuatan Tugas Akhir	2
1.5 Manfaat Tugas Akhir	2
1.6 Metodologi	3
1.7 Sistematika	5
BAB II TINJAUAN PUSTAKA	7
2.1 Intel Realsense	7
2.2 Avatar	9
2.3 Blender 3D	10
BAB III ANALISIS DAN PERANCANGAN	12
3.1 Analisis Sistem	13
3.2 Perancangan UI (User Interface) <i>Faceshift</i>	13
3.3 Perancangan Algoritma	14
3.3.1 Rancangan Pendeteksi ekspresi Wajah	14
3.3.2 Rancangan Pergerakan Animasi Karakter	15

3.3.3	Rancangan Pengganti karakter.....	18
3.4	Perancangan Perangkat Lunak	19
3.4.1	Deskripsi Umum Perangkat Lunak.....	19
3.4.2	Spesifikasi Kebutuhan Fungsional	19
3.4.3	Spesifikasi Kebutuhan Non Fungsional.....	20
3.4.4	Karakteristik Pengguna.....	20
3.5	Perancangan Sistem.....	21
3.5.1	Mendeteksi wajah pengguna.....	22
3.5.2	Mendeteksi dan menampilkan emosi pengguna ...	23
3.5.3	Mengganti karakter.....	24
3.5.4	Menampilkan petunjuk	25
BAB IV	IMPLEMENTASI	27
4.1	Lingkungan Implementasi.....	27
4.2	Implementasi Pembuatan Scene.....	28
4.3	Implementasi Pembuatan Script.....	30
4.4	Implementasi Tampilan Antarmuka Menu.....	31
4.5	Implementasi Mendeteksi Wajah	34
4.6	Implementasi pendeteksi Emosi Pengguna	40
4.7	Implementasi Menggerakan Animasi Karakter.....	44
4.8	Implementasi Ganti Karakter	52
BAB V	PENGUJIAN DAN EVALUASI.....	57
5.1	Lingkungan Uji Coba	57
5.2	Lingkungan Perangkat Keras	57
5.3.	Lingkungan Perangkat Lunak	57

5.4	Pengujian Fungsionalitas.....	58
5.4.1	Skenario Pengujian Mendeteksi Wajah	58
5.4.2	Skenario Pengujian Menggerakan Kepala	59
5.4.3	Skenario Pengujian Mendeteksi Emosi.....	63
5.4.4	Seknario Pengujian Mengganti Karater	69
5.4.5	Hasil Pengujian Fungsional	70
5.5	Pengujian Pengguna	71
5.5.1	Hasil Penilaian	71
5.5.2	Tanggapan pengguna	72
5.5.3	Evaluasi pengujian	73
BAB VI KESIMPULAN DAN SARAN		75
6.1	Kesimpulan.....	75
6.2	Saran.....	75
DAFTAR PUSTAKA		77
LAMPIRAN.....		79

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

Tabel 3.1 Karakteristik pengguna	21
Tabel 3.2 Spesifikasi kasus penggunaan UC-001	22
Tabel 3.3 Spesifikasi kasus penggunaan UC-002	23
Tabel 3.4 Spesifikasi kasus penggunaan UC-003	24
Tabel 3.5 Spesifikasi kasus penggunaan UC-004	25
Tabel 4.1 Lingkupan Implementasi Perangkat Lunak	27
Tabel 5.1 Tabel perangkat keras	57
Tabel 5.2 Tabel perangkat lunak	58
Tabel 5.3 Ujicoba UC-001	58
Tabel 5.4 Tabel Ujicoba UC-002	60
Tabel 5.5 Tabel Ujicoba UC-003	61
Tabel 5.6 Tabel Ujicoba UC-004	62
Tabel 5.7 Tabel Ujicoba UC-005	63
Tabel 5.8 Tabel Ujicoba UC-006	64
Tabel 5.9 Tabel Ujicoba UC-007	65
Tabel 5.10 Tabel Ujicoba UC-008	66
Tabel 5.11 Tabel Ujicoba UC-009	67
Tabel 5.12 Tabel Ujicoba UC-010	68
Tabel 5.13 Ujicoba UC-011	69
Tabel 5.14 Hasil rekapitulasi	70
Tabel 5.15 Tabel nilai pengujian	71
Tabel 5.16 Hasil Tabel quisioner	72
Tabel 5.17 Tabel Tanggapan	72

(Halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

Gambar 2.1 Perangkat <i>Intel Realsense</i>	7
Gambar 2.2 Komponen pada perangkat Intel Realsense	8
Gambar 2.3 Kamera Inframerah pada intel realsense	Error!
Bookmark not defined.	
Gambar 2.4 Komponen resonant micromirror pada perangkat Intel Realsense.	9
Gambar 2.5 Gambar Kegunaan UX	10
Gambar 2.6 Tampilan Blender 3D	11
Gambar 3.1 flowchart pendeteksi emosi	16
Gambar 3.2 flowchart pergerakan karakter	17
Gambar 3.3 flowchart ganti karakter	18
Gambar 3.4 Diagram kasus	22
Gambar 4.1 Penambahan scene	28
Gambar 4.2 Tampilan scene baru	29
Gambar 4.3 Tampilan menu scene	29
Gambar 4.4 Pembuatan Script (1)	30
Gambar 4.5 Pembuatan Script (2)	30
Gambar 4.6 Pembuatan <i>script</i> (3)	31
Gambar 4.7 Antarmuka Menu	32
Gambar 4.8 Antarmuka petunjuk	33
Gambar 4.9 Antarmuka utama	34
Gambar 4.10 Antarmuka utama	40
Gambar 4.11 Memasukan data emotion dan sentiment	41
Gambar 4.12 Tampilan emosi pada aplikasi	44
Gambar 4.13 Tampilan nilai lokasi objek pada karakter	46
Gambar 4.14 Tampilan Karakter sebelum bergerak	52
Gambar 4.15 Tampilan Karakter sesudah bergerak	52
Gambar 4.16 Tampilan Karakter sebelum diganti	55
Gambar 4.17 Tampilan Karakter sesudah diganti	55

Gambar 5.1 Hasil pengujian mendeteksi wajah pengguna	59
Gambar 5.2 Gambar implementasi yawl.....	60
Gambar 5.3 Gambar Implementasi <i>roll</i>	61
Gambar 5.4 Gambar Implementasi <i>roll</i>	62
Gambar 5.5 Gambar implementasi happy.....	63
Gambar 5.6 Gambar implementasi Sad	64
Gambar 5.7 Gambar implementasi Angry	65
Gambar 5.8 Gambar implementasi Sureprice	66
Gambar 5.9 Gambar implementasi Fear	67
Gambar 5.10 Gambar implementasi Disgust	68
Gambar 5.11 Hasil ujicoba sebelum mengganti karakter	69
Gambar 5.12 Hasil ujicoba setelah diganti karakter	70

DAFTAR KODE

Kode 4.1 tombol main dan keluar	32
Kode 4.2 Antarmuka petunjuk	33
Kode 4.3 kode validasi data wajah.....	36
Kode 4.4 ambil data distance pada wajah	36
Kode 4.5 kode mirror pada kamera.....	37
Kode 4.6 kode menyimpan ekspresi wajah.....	38
Kode 4.7 menyimpan pergerakan kepala	38
Kode 4.8 Kode menyimpan pergerakan badan	39
Kode 4.9 Kode menampilkan landmark pada gambar wajah pengguna	40
Kode 4.10 Kode mendapatkan emotion dari query emotion ..	42
Kode 4.11 Kode mendapatkan sentiment dan emotion tertinggi	43
Kode 4.12 Kode menyimpan data emotion dan sentiment	44
Kode 4.13 Kode Insisiasi value bagian dari objek karakter....	45
Kode 4.14 Kode inisiasi indeks pergerakan objek animasi	47
Kode 4.15 Kode Sumber Fungsi untuk mengecek karakter....	48
Kode 4.16 Kode Sumber nilai dari facetracking	49
Kode 4.17 Kode Sumber value yang digunakan untuk menerima data facetracking kedalam data indeks karakter	51
Kode 4.18 Kode Sumber untuk menggerakkan karakter dari karakter sesuai data dari indeks karakter	52
Kode 4.19 kode ganti karakter	54

(Halaman ini sengaja dikosongkan)

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan olah citra digital telah mengalami peningkatan yang sangat signifikan pada awal tahun 2010. Sebelumnya, citra digital khususnya animasi komputer pertama kali hanya sekadar sebuah garis dan bulat, akan tetapi sekarang animasi telah berbentuk hamper seperti asli. Pada awalnya, animasi komputer digunakan dengan tujuan penelitian ilmiah, teknik dan ilmu pengetahuan lain nya. Dengan majunya teknologi komputer, semakin canggih hasil yang dihasilkan dari animasi komputer sehingga animasi komputer juga digunakan untuk tujuan hiburan dan lain-lain.

Untuk membuat animasi pun beragam, seperti dengan cara manual, dynamic simulation dan motion capture. Motion capture adalah metode pembuatan animasi dengan menggunakan gerakan asli, kemudian ditangkap dengan alat motion capture dan hasilnya kemudian diaplikasikan ke objek digital [1]. Akan tetapi banyak alat untuk menangkap motion pada tubuh, khususnya pada muka yang mengharuskan pengguna untuk menggunakan helm khusus dan marker pada muka untuk mendeteksi ekspresi yang dilakukan oleh pengguna.

Intel Realsense adalah pengembangan teknologi dari Intel, dimana Realsense menempatkan diri di teknologi tersebut sama seperti kompetitor lainnya yang menjadi teknologi awalnya yakni Kinect oleh primesense, dan Leap Motion oleh Leap. Fitur yang dimiliki oleh Intel Realsense sendiri adalah dapat melakukan Facial Analysis seperti deteksi wajah, mata, ekspresi, Hand & Finger Tracking dimana kita bisa mendeteksi setiap sendi yang ada di jari kita, speech recognition dimana kita bisa mengenali suara dan bisa mengendalikan segala sesuatu dikomputer dengan suara, background subtraction, dan Augmented Reality [2].

Pada tugas akhir ini akan mencoba menerapkan fitur pada teknologi Intel RealSense yaitu Face Tracking untuk membuat ekspresi pada karakter avatar sehingga dihasilkan ekspresi wajah yang terlihat nyata secara real time.

1.2 Rumusan Masalah

Berdasarkan uraian pada latar belakang dapat dirumuskan masalah dalam tugas akhir adalah sebagai berikut:

1. Bagaimana cara mengimplementasikan teknologi *Landmark Position Intel Realsense* untuk membuat aplikasi *faceshift*?
2. Bagaimana cara menggabungkan teknologi *Landmark Position* dengan *Emotion Recognition* untuk membuat aplikasi *faceshift*?
3. Bagaimana cara mengimplementasikan pergantian karakter pada aplikasi

1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan antara lain:

1. Aplikasi yang dibuat merupakan aplikasi desktop.
2. Lingkungan pengembangan yang digunakan menggunakan aplikasi Unity 3D dan bahasa pemrograman C#.
3. Aplikasi ini menggunakan Intel Realsense.
4. Menggunakan Intel Realsense Camera.
5. Aplikasi hanya bisa berjalan ketika perangkat komputer telah terpasang Intel Realsense.
6. Emosi yang dapat ditampilkan oleh *intel realsense* adalah 6.

1.4 Tujuan Pembuatan Tugas Akhir

Tujuan dari pembuatan tugas akhir ini adalah untuk membuat aplikasi *faceshift* dengan bantuan teknologi *Intel Realsense*.

1.5 Manfaat Tugas Akhir

Manfaat dari hasil pembuatan tugas akhir ini antara lain:

1. Memudahkan pembuatan animasi khususnya pada pergerakan muka agar dapat menghasilkan ekspresi muka yang nyata
2. Mengenalkan teknologi *Intel Realsense* salah satunya penggunaan fitur *face tracking* kepada masyarakat luas.

1.6 Metodologi

Pembuatan tugas akhir dilakukan menggunakan metodologi sebagai berikut:

A. Studi literatur

Tahap studi literatur merupakan tahap pembelajaran dan pengumpulan informasi yang digunakan untuk implementasi tugas akhir. Tahap ini diawali dengan pengumpulan literatur, diskusi, eksplorasi teknologi, dan pustaka, serta pemahaman dasar teori yang digunakan pada topik tugas akhir. Literatur-literatur yang dimaksud disebutkan sebagai berikut:

1. *Intel Realsense*.
2. SDK Intel Realsense.
3. Blender 3D.
4. Avatar.
5. Unity.

B. Perancangan Perangkat Lunak

Pada tahap ini diawali dengan melakukan analisis awal terhadap permasalahan utama yang muncul pada topik tugas akhir. Kemudian dilakukan perancangan perangkat lunak yang meliputi penentuan data yang akan digunakan dan proses-proses yang akan dilaksanakan. Langkah yang akan digunakan pada tahap ini adalah sebagai berikut:

1. Perancangan pendeteksi muka (*face detection*).
2. Perancangan pergerakan avatar (*face animation*).
3. Perancangan fitur-fitur tambahan seperti pengaturan dasar aplikasi dan penyesuaian tata letak.

C. Implementasi dan pembuatan sistem

Pada tahap ini dilakukan membuat karakter dan latar dengan menggunakan aplikasi *Blender 3D*. Aplikasi ini dibangun menggunakan aplikasi *Unity* dengan bahasa dasar *C#* dan

Javascript. Agar aplikasi ini dapat mendeteksi dengan perangkat *Intel Realsense* maka terlebih dahulu *PC (Personal Computer)* harus terinstall driver *Intel Realsense*.

D. Uji coba dan evaluasi

Pada tahap ini dilakukan uji coba dengan menggunakan beberapa macam kondisi untuk aplikasi bisa berjalan atau tidak. Uji fungsionalitas untuk mengetahui apakah aplikasi sudah memenuhi semua kebutuhan fungsional.

Pengujian aplikasi atau sistem dilakukan untuk mengukur suatu keberhasilan aplikasi ini dengan meminimalisir persepsi dan kesalahan-kesalahan. Pengujian aplikasi dengan cara menggunakan metode kuesioner.

Kuesioner ini akan dibagikan kepada pengguna setelah mengoperasikan aplikasi drum virtual ini dan membandingkan hasil antara desain satu dengan desain yang lainnya. Selain itu untuk menganalisis kesalahan-kesalahan pengguna saat menggunakan aplikasi *faceshift*, yang harus dilakukan adalah merekam layar aplikasi kemudian dicari kesalahan-kesalahan yang terjadi. Apabila hasil pengujian tersebut kurang memuaskan, maka yang harus dilakukan adalah memperbaiki aplikasi baik dari sisi desain ataupun program.

E. Penyusunan laporan tugas akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

1. Pendahuluan

- a. Latar Belakang
- b. Rumusan Masalah
- c. Batasan Tugas Akhir
- d. Tujuan
- e. Metodologi
- f. Sistematika Penulisan

2. Tinjauan Pustaka

3. Analisis dan Perancangan
4. Implementasi
5. Pengujian dan Evaluasi
6. Penutup

1.7 Sistematika

Buku tugas akhir ini terdiri dari beberapa bab yang dijelaskan sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan dan batasan permasalahan, tujuan dan manfaat pembuatan tugas akhir, metodologi yang digunakan, dan sistematika penyusunan tugas akhir.

BAB II TINJAUAN PUSTAKA

Bab ini membahas dasar pembuatan dan beberapa teori penunjang yang berhubungan dengan pokok pembahasan yang mendasari pembuatan tugas akhir ini.

BAB III ANALISIS DAN PERANCANGAN

Bab ini membahas analisis dari sistem yang dibuat meliputi analisis permasalahan, deskripsi umum perangkat lunak, spesifikasi kebutuhan, dan identifikasi pengguna. Kemudian membahas rancangan dari sistem yang dibuat meliputi rancangan skenario kasus penggunaan, arsitektur, data, dan antarmuka.

BAB IV IMPLEMENTASI

Bab ini membahas implementasi dari rancangan sistem yang dilakukan pada tahap perancangan. Penjelasan implementasi meliputi implementasi pembuatan simulasi ibadah haji dengan menggunakan Oculus Rift, dan antar muka aplikasi.

BAB V**PENGUJIAN DAN EVALUASI**

Bab ini membahas pengujian dari aplikasi yang dibuat dengan melihat keluaran yang dihasilkan oleh aplikasi dan evaluasi untuk mengetahui kemampuan aplikasi.

BAB VI**PENUTUP**

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan serta saran untuk pengembangan aplikasi selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Intel Realsense

Intel Realsense 3D Camera adalah teknologi kamera yang mampu merespon tangan, lengan, dan gerakan kepala serta ekspresi wajah. kemampuan kamera *Intel Realsense* ini dapat mendeteksi 22 sendi di satu telapak tangan kita dan juga bisa mendeteksi kedalaman 3 Dimensi. Bentuk kamera *intel realsense* dapat dilihat pada Gambar 2.1:

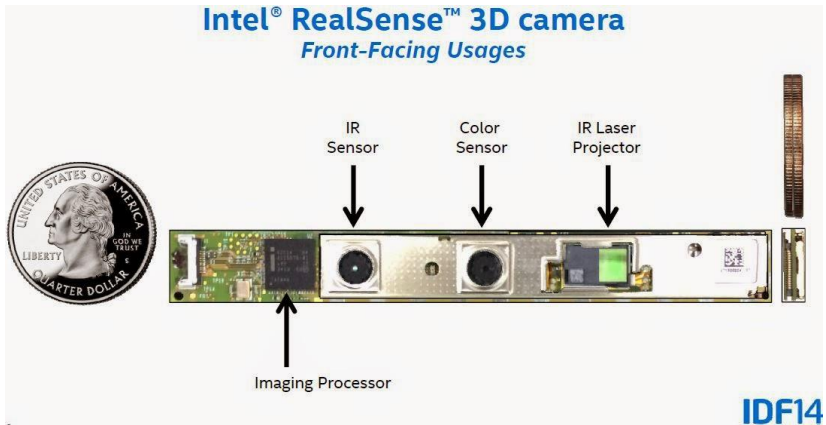


Gambar 2.1 Perangkat *Intel Realsense*

Sumber. <http://intelrealsense.bemyapp.com/>

Rahasia dari kecanggihan teknologi Intel Realsense adalah perangkat dengan kamera Intel RealSense 3D memiliki tiga lensa. Jadi terdapat 3 kamera yang saling terintegrasi, tiga kamera tersebut adalah

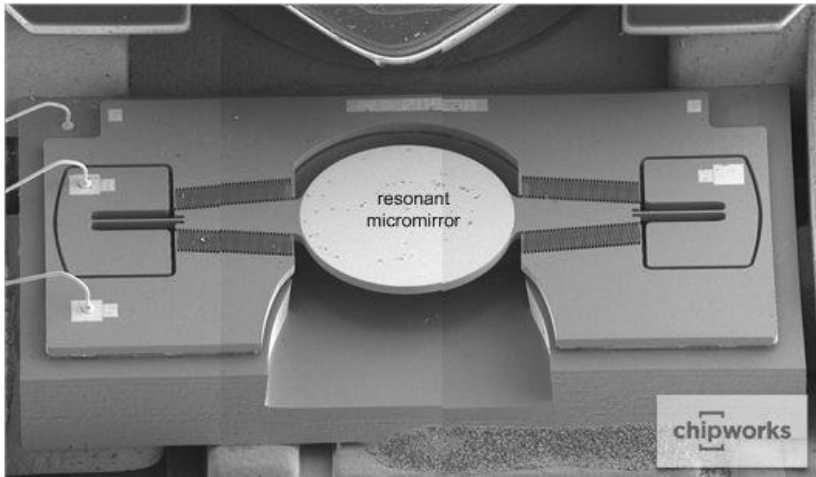
kamera konvensional, kamera inframerah, dan laser proyektor inframerah. Ketiga lensa kamera tersebut memungkinkan perangkat untuk menyimpulkan kedalaman dengan mendeteksi cahaya inframerah dari benda-benda yang ada di depannya. Kemudian data visual yang diambil dikombinasikan dengan software pelacak gerak Intel RealSense [3]. Sensor kamera dapat dilihat pada Gambar 2.2:



Gambar 2.2 Komponen pada perangkat Intel Realsense

Sumber: http://education.rec.ri.cmu.edu/content/electronics/boe/ir_sensor/1.html

Resonan pada IR Laser Projector dapat dilihat pada Gambar



Gambar 2.3 Komponen resonant micromirror pada perangkat Intel Realsense.

Sumber: <http://www.chipworks.com/about-chipworks/overview/blog/inside-the-intel-realsense-gesture-camera>

2.2 Avatar

Film animasi, atau biasa disingkat animasi saja, adalah film yang merupakan hasil dari pengolahan gambar tangan sehingga menjadi gambar yang bergerak. Pada awal penemuannya, film animasi dibuat dari berlembar-lembar kertas gambar yang kemudian di-"putar" sehingga muncul efek gambar bergerak. Dengan bantuan komputer dan grafika komputer, pembuatan film animasi menjadi sangat mudah dan cepat. Bahkan akhir-akhir ini lebih banyak bermunculan film

animasi 3 dimensi daripada film animasi 2 dimensi. Bentuk karakter dapat dilihat pada Gambar 2.5:



Gambar 2.4 Gambar Kegunaan UX

Sumber: <http://animation.binus.ac.id/files/2014/03/cupcake03.jpg>

Avatar merupakan karakter yang ada di dalam film animasi. Avatar terdiri dari 2D dan 3D. Pembuatan avatar juga bermacam macam salah satunya adalah dengan cara manual atau dengan komputer. Untuk manual dan komputer sang creator memerlukan contoh dari orang atau hewan nyata agar dapat membentuk avatar yang nyata. Akan tetapi untuk pembuatan dengan komputer dapat lebih mudah karena tidak perlunya membuat setiap frame dari avatar. Pembuatan ekspresi karakter dengan komputer bisa dengan manual atau dapat otomatis dengan bantuan alat, salah satunya adalah kamera *Intel Realsense*.

2.3 Blender 3D

Blender adalah perangkat lunak sumber terbuka grafika komputer 3D. Perangkat lunak ini digunakan untuk membuat film animasi, efek visual, model cetak 3D, aplikasi 3D interaktif dan permainan video. Blender memiliki beberapa fitur termasuk pemodelan 3D, penteksturan, penyunting gambar bitmap, penulangan,

simulasi cairan dan asap, simulasi partikel, animasi, penyunting video, pemahat digital, dan *rendering*. Tampilan antarmuka Blender 3D pada Gambar 2.6:



Gambar 2.5 Tampilan Blender 3D

Sumber: <http://www.cgmeetup.net/home/uv-unwrapping-and-texture-painting-in-blender-tutorial/>

(Halaman ini sengaja dikosongkan)

BAB III

ANALISIS DAN PERANCANGAN

Bab ini menjelaskan tentang analisis dan perancangan aplikasi drum virtual berbasis teknologi *Intel Realsense*. Pembahasan yang akan dilakukan meliputi analisis fitur yang dibutuhkan dan perancangan perangkat lunak.

3.1 Analisis Sistem

Aplikasi ini dibangun untuk memudahkan creator animasi membuat karakternya agar terlihat lebih hidup. Karena pembuatan animasi wajah yang bagus dan cepat membutuhkan peralatan lain yang lebih mahal seperti *helmet camera rig* atau creator dapat menggunakan aplikasi saja dan menghasilkan ekspresi yang kurang hidup dan cenderung memakan waktu yang lama.

Aplikasi ini dibangun menggunakan teknologi Intel Realsense sehingga pengguna dapat membuat ekspresi karakter sesuai dengan ekspresi dari pengguna itu sendiri. Aplikasi ini dibangun menggunakan Unity karena fitur-fitur pada aplikasi ini tergolong lengkap dan mudah untuk dipahami. Fitur tersebut seperti pembuatan kamera, pendeteksi wajah, mengganti karakter, deteksi emosional dengan imk *UI (User Interface)* serta *UX (User Experience)* yang baik.

3.2 Perancangan UI (User Interface) *Faceshift*

Perancangan *UI (User Interface)* aplikasi *faceshift* menggunakan *Blender 3D*. Pembuatan desain karakter dan latar dibangun dengan *Blender 3D*. Desain karakter dan latar yang dibuat merupakan desain 3D. Aplikasi ini termasuk open source yang fiturnya cukup lengkap serta banyak animator yang membuat karakter menggunakan aplikasi ini. Setelah itu diberikan pencahayaan pada unity agar memberikan kesan yang lebih nyata.

3.3 Perancangan Algoritma

3.3.1 Rancangan Pendeteksi ekspresi Wajah

Ketika titik-titik pada wajah berhasil ditampilkan di layar dengan fitur *face tracking* pada *Intel Realsense*, maka tahapan selanjutnya adalah mengolah data tersebut agar didapatkan ekspresi pada wajah.

Untuk mendeteksi ekspresi yang sedang dilakukan oleh pengguna, *Intel Realsense* menyediakan *library Emotion Detection* yang digunakan untuk mendeteksi emosi dari pengguna. *Library Emotion Detection* menyediakan 2 parameter yaitu *evidence* dan *intensity* [3]. *Evidence* adalah nilai yang merepresentasikan kemungkinan di dalam berbasis 10 bahwa ekspresi tersebut terjadi dalam waktu sekarang. Sebagai contoh ekspresi senyum yang dilakukan pengguna memiliki nilai 2 untuk ekspresi *Happy* sehingga mengidentifikasi bahwa nilai tersebut memiliki 100 kali lebih terkatagoris pada ekspresi *happy*, dan nilai -2 pada ekspresi *Sad* mengidentikasi bahwa nilai tersebut memiliki 100 kali kemungkinan tidak terjadi pada ekspresi *sad*. *Intensity* adalah nilai yang menunjukkan intensitas suatu ekspresi muncul dengan nilai pada Tabel 3.1.

Tabel 4.1 Interval Intensity

Value Interval	Description of Emotion Expression
0.0 , 0,2	Expression is likely absent
0.2 , 0,4	Expression is of low intensity
0.4 , 0,6	Expression is of medium intensity
0.6 , 0,8	Expression is of high intensity
0.8 , 1,0	Expression is of high intensity

Library Emotion Detection memiliki 6 data ekspresi yaitu *happy*, *sad*, *angry*, *surprise*, *disgust* dan *fear*, serta 3 sentimen positive negative dan neutral yang memiliki nilai *intensity* dan *evidence* yang berbeda sehingga *intel realsense* dengan otomatis mengetahui ekspresi pengguna dengan memasukan nilai *landmark* pengguna kedalam tiap

ekspresi dan sentimen. Setelah itu nilai dari masing masing ekspresi dan sentiment diambil paling tinggi untuk diketahui ekspresi mana yang cocok pada pengguna. Contohnya ketika pengguna senyum, maka nilai *intencity* dan *evidence* pada *happy* akan lebih besar dibandingkan ekspresi yang lain sehingga ekspresi yang dilakukan adalah *happy*. Untuk mendeteksi ekspresi pertama kali diperlukan adalah memasukan 6 data ekspresi dan sentiment pada wajah kedalam *mainFace*. Kemudian ekspresi pengguna direkam dan di cari *intencity* dan *evidence* pada setiap 6 data ekspresi dan 3 data sentiment. Setiap ekspresi akan memberikan nilai *intencity* dan *evidence* yang berbeda sesuai dengan jenis ekspresi dan sentiment 6 tipe emosi yang telah disediakan oleh *library intel realsense*.

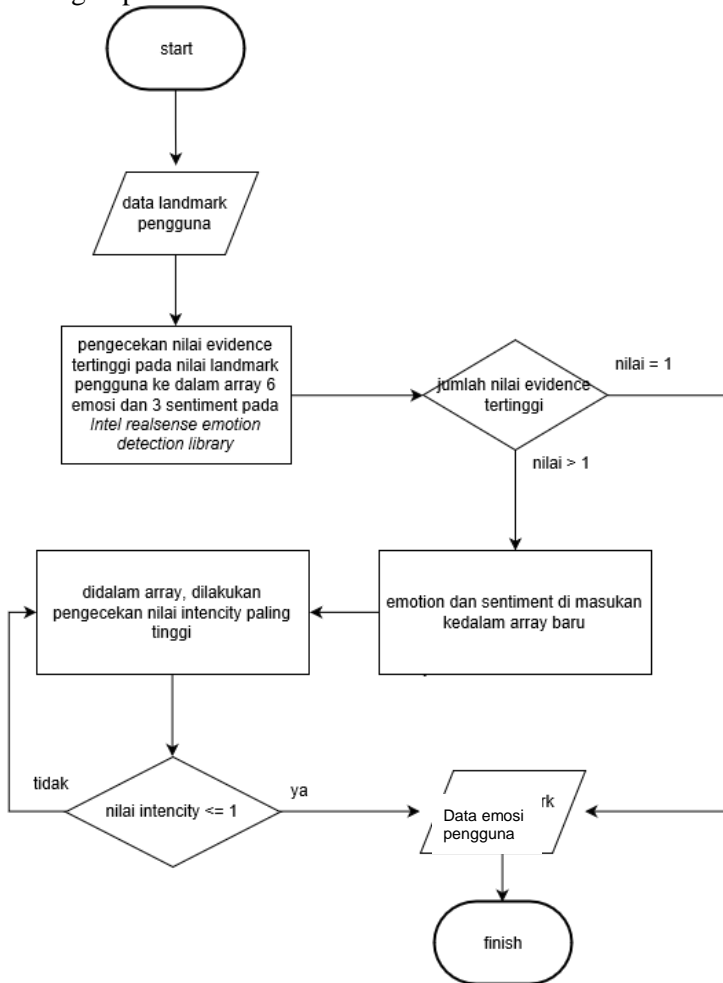
Terkadang nilai ekspresi atau sentiment akan memiliki nilai yang sama pada *evidence* nya. Contoh apabila ekspresi pengguna adalah diam dan sedikit sedih maka system akan mendeteksi neutral dan negative akan bernilai 1. Oleh karena itu dicari nilai *evidence* paling tinggi untuk mendapatkan ekspresi dan sentiment yang benar. Apabila nilai *evidence* tertinggi adalah satu ekspresi dan satu sentiment maka ekspresi tersebut merupakan ekspresi dari pengguna.

Akan tetapi apabila nilai *evidence* dari salah satu yang sama, maka nilai tersebut dicari *intencity* tertinggi dari keduanya. Proses lebih lanjut akan dijelaskan pada Gambar flow chart 3.4.

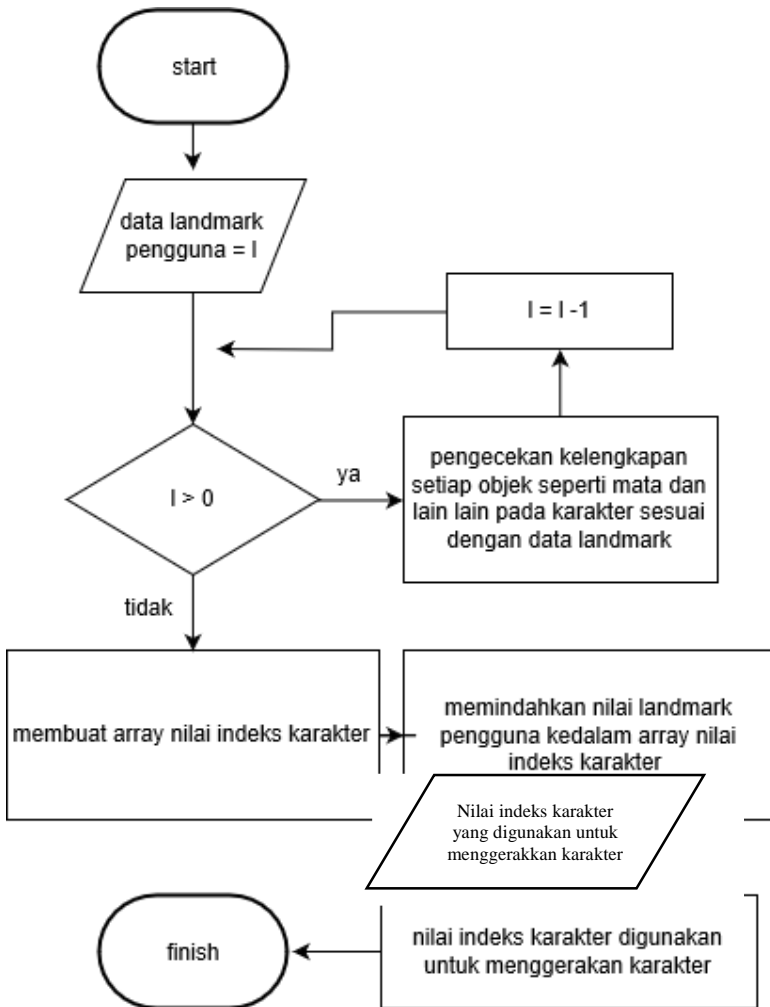
3.3.2 Rancangan Pergerakan Animasi Wajah Karakter

Untuk mengerakkan wajah dari karakter, pertama kali di inisialisasi object pada karakter seperti *bodyobject*, *mouthobject*, dan *eyeobject*. *Intel Realsense SDK* memiliki *library faceanimation* untuk pergerakan ekspresi wajah seperti *leftEyeRotation*, *rightEyeRotation*, *jawposition* dan *gesture* lainnya sesuai dengan object masing masing. Untuk menggerakan ekspresi tersebut terdapat 2 data yang berbeda, yang pertama digunakan untuk menggerakan karakter, dan yang kedua data *landmark* pengguna. Contohnya *_rBrowDownInitialValue* digunakan untuk menyimpan nilai pergerakan karakter dan *_rBrowDownBodyBlendShapeIndex* digunakan untuk menyimpan data dari wajah pengguna. Kemudian dicek apakah karakter memiliki

objek yang diperlukan atau tidak. Apabila karakter memiliki semua objek yang diperlukan, maka data dari *face tracking* dipindahkan untuk menggerakkan karakter sesuai dengan gerakan wajah pengguna. Berikut adalah flowchart animasi wajah karakter pada aplikasi yang akan dibangun pada Gambar 3.5.



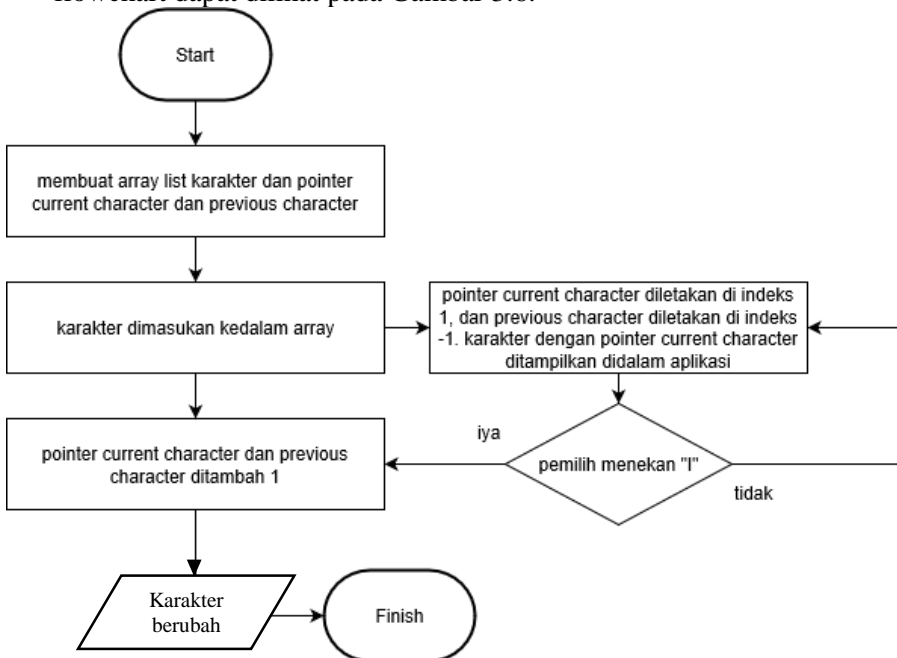
Gambar 4.1 flowchart pendeteksi emosi



Gambar 4.2 flowchart pergerakan karakte

3.3.3 Rancangan Pengganti karakter

Aplikasi yang akan dibangun akan memiliki fitur tambahan yaitu pengguna dapat mengganti karakter yang digunakan sesuai dengan jumlah karakter yang ada di aplikasi. Untuk mengganti karakter pertama kali semua karakter dimasukkan ke dalam array karakter. Setelah itu di set urutan dari semua karakter. Untuk karakter pertama diset sebagai *current character*. Ketika karakter diganti, *current character* dipindahkan ke urutan setelahnya dan *enviromtent* dan *object* dipindahkan ke depan kamera dan *setactive* dihidupkan.. Karakter sebelumnya diambil *enviromtent* dan *object* kemudian dipindahkan ke tempat yang lain dan *setactive* dimatikan. Gambar flowchart dapat dilihat pada Gambar 3.6.



Gambar 4.3 flowchart ganti karakter

3.4 Perancangan Perangkat Lunak

Pada subbab ini akan dibahas mengenai deskripsi umum perangkat lunak, spesifikasi kebutuhan fungsional dan kebutuhan non-fungsional serta bagaimana karakteristik pengguna aplikasi.

3.4.1 Deskripsi Umum Perangkat Lunak

Aplikasi yang akan dibangun adalah sebuah aplikasi yang menggerakkan ekspresi karakter animasi. Aplikasi *faceshift* ini akan menampilkan badan manusia secara penuh seperti kepala, badan dan tangan. Sudut pandang yang digunakan pada aplikasi ini adalah mengambil dari sisi depan.

Cara memainkan aplikasi ini menggunakan pergerakan wajah. Tidak seperti aplikasi animasi lain nya, aplikasi ini menggunakan wajah pengguna digunakan untuk menggerakkan pergerakan wajah karakter secara langsung. Pergerakan dimaksudkan adalah pergerakan kepala, badan dan ekspresi wajah pada karakter menggunakan kamera *Intel Realsense*.

3.4.2 Spesifikasi Kebutuhan Fungsional

Berdasarkan deskripsi umum sistem, maka disimpulkan bahwa kebutuhan fungsional dari aplikasi ini yaitu:

1. Menggerakkan karakter animasi
Aplikasi ini mempunyai fungsionalitas utama yaitu menggerakkan badan dan wajah karakter melalui *input* yang telah diterima.
2. Mendeteksi dan menampilkan emosi
Aplikasi ini dapat mendeteksi dan menampilkan kondisi perasaan pengguna yang disampaikan melalui pergerakan wajah.
3. Mengganti karakter

Aplikasi ini dapat mengganti karakter yang berbeda

4. Menampilkan petunjuk

Aplikasi ini dapat menampilkan petunjuk penggunaan aplikasi.

3.4.3 Spesifikasi Kebutuhan Non Fungsional

Terdapat kebutuhan non-fungsional yang apabila dipenuhi, dapat meningkatkan kualitas dari aplikasi ini. Berikut daftar kebutuhan non-fungsional:

1. Aspek performa

Aspek performa ini meliputi waktu jeda input dari pengguna dengan reaksi yang diberikan oleh aplikasi. Semakin cepat waktu jeda maka kenyamanan penggunaan aplikasi semakin meningkat. Dengan kata lain, aplikasi dapat berjalan secara realtime. Aplikasi maksimal harus merespon maksimal 1 detik.

2. Kebutuhan grafis

Daya tarik aplikasi berbanding lurus dengan kualitas grafis yang disajikan dalam aplikasi. Semakin bagus desain aplikasi maka semakin besar daya tarik pengguna untuk menggunakan aplikasi ini. Selain itu pengguna akan tertarik menggunakan aplikasi jika tampilan atau desain pada aplikasi menarik.

3. Aspek operasional

Aspek operasional yang mempengaruhi aplikasi ini dengan lingkungan pengoperasional, yaitu meliputi jumlah *core* CPU perangkat, RAM, kecepatan processor dan lain lain.

3.4.4 Karakteristik Pengguna

Pengguna yang akan menggunakan aplikasi ini berjumlah satu orang, yaitu pengguna yang akan melakukan simulasi atau pembelajaran. Karakteristik pengguna tercantum dalam Tabel 3.1.

Tabel 4.2 Karakteristik pengguna

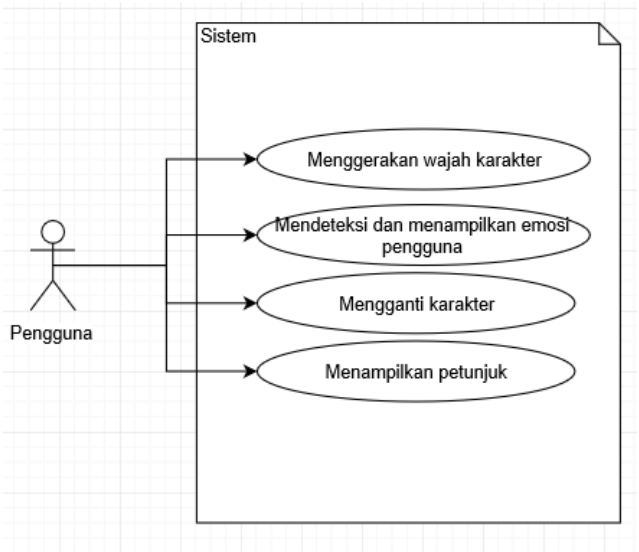
Nama Aktor	Tugas	Hak Akses Aplikasi	Kemampuan yang harus dimiliki
Pengguna	Pihak luar yang mencoba aplikasi	Menjalankan aplikasi	Tidak ada

3.5 Perancangan Sistem

Tahap perancangan dalam subbab ini dibagi menjadi beberapa bagian yaitu perancangan diagram kasus penggunaan, perancangan skenario kasus penggunaan, perancangan arsitektur, perancangan antarmuka pengguna, dan perancangan kontrol aplikasi.

Dalam aplikasi tugas akhir ini, terdapat 4 kasus penggunaan yang ada yaitu dapat mendeteksi wajah pengguna, mendeteksi dan menampilkan emosi pengguna, menggerakkan wajah karakter, dan mengganti karakter. Diagram kasus untuk tugas akhir ini dapat dilihat pada Gambar 3.7.

Setelah diketahui bagaimana perancangan kasus penggunaan maka spesifikasi kasus penggunaan akan dijelaskan pada subbab berikut:



Gambar 4.4 Diagram kasus

3.5.1 Mendeteksi wajah pengguna

Spesifikasi kasus Mendeteksi wajah pengguna dapat dijabarkan pada Tabel 3.2.

Tabel 4.3 Spesifikasi kasus penggunaan UC-001

Nama Kasus Penggunaan	Mendeteksi wajah pengguna
Kode	UC-001
Deskripsi	Kasus penggunaan agar wajah actor dapat menggerakan wajah karakter secara langsung
Aktor	Pengguna
Kondisi Awal	Pengguna sudah membuka aplikasi

Nama Kasus Penggunaan	Mendeteksi wajah pengguna
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna menyambungkan kamera <i>Intel Realsense</i> dengan perangkat. 2. Pengguna membuka aplikasi 3. Sistem menampilkan karakter 3D pada monitor 4. Pengguna dapat menggerakkan aplikasi dengan menggerakkan kepala dan mengubah ekspresi wajah. 5. Aplikasi menggerakkan wajah karakter sesuai dengan pergerakan pengguna 6. Pengguna dapat menampilkan <i>landmark</i> yang sedang digunakan pada wajah pengguna 7. Sistem menampilkan posisi <i>landmark</i> pada monitor
Alur Kejadian Alternatif	-

3.5.2 Mendeteksi dan menampilkan emosi pengguna

Spesifikasi kasus menampilkan emosi pengguna dapat dijabarkan pada Tabel 3.3.

Tabel 4.4 Spesifikasi kasus penggunaan UC-002

Nama Kasus Penggunaan	Mendeteksi dan menampilkan emosi pengguna
Kode	UC-002

Nama Kasus Penggunaan	Mendeteksi dan menampilkan emosi pengguna
Deskripsi	Kasus penggunaan agar aktor dapat menampilkan emosi dari pengguna.
Aktor	Pengguna
Kondisi Awal	Pengguna sudah membuka aplikasi
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna mengganti ekspresi wajah 2. Sistem mendeteksi perubahan ekspresi kemudian mengidentifikasi ekspresi tersebut 3. Sistem menampilkan jenis ekspresi pada monitor.
Alur Kejadian Alternatif	-

3.5.3 Mengganti karakter

Spesifikasi kasus Mengganti Karakter dapat dijabarkan pada Tabel 3.4.

Tabel 4.5 Spesifikasi kasus penggunaan UC-003

Nama Kasus Penggunaan	Mengganti karakter
Kode	UC-003
Deskripsi	Kasus penggunaan agar aktor dapat mengganti karakter yang digunakan
Aktor	Pengguna
Kondisi Awal	Pengguna berada pada menu awal pada halaman utama

Nama Kasus Penggunaan	Mengganti karakter
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna memilih ganti karakter 2. Sistem akan mengganti karakter
Alur Kejadian Alternatif	-

3.5.4 Menampilkan petunjuk

Spesifikasi kasus Mengganti Karakter dapat dijabarkan pada Tabel 3.5.

Tabel 4.6 Spesifikasi kasus penggunaan UC-004

Nama Kasus Penggunaan	Menampilkan petunjuk
Kode	UC-004
Deskripsi	Kasus penggunaan agar aktor dapat melihat petunjuk cara pemakaian kamera <i>Intel Realsense</i> dalam aplikasi.
Aktor	Pengguna
Kondisi Awal	Pengguna berada pada menu awal pada halaman muka
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna memilih tombol main 2. Sistem menampilkan petunjuk cara pemakaian
Alur Kejadian Alternatif	-

(Halaman ini sengaja dikosongkan)

BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi dari perancangan perangkat lunak. Di dalamnya mencakup proses penerapan dan pengimplementasian algoritma dan antarmuka yang mengacu pada rancangan yang telah dibahas sebelumnya.

4.1 Lingkungan Implementasi

Lingkungan implementasi pada saat mengerjakan tugas akhir.

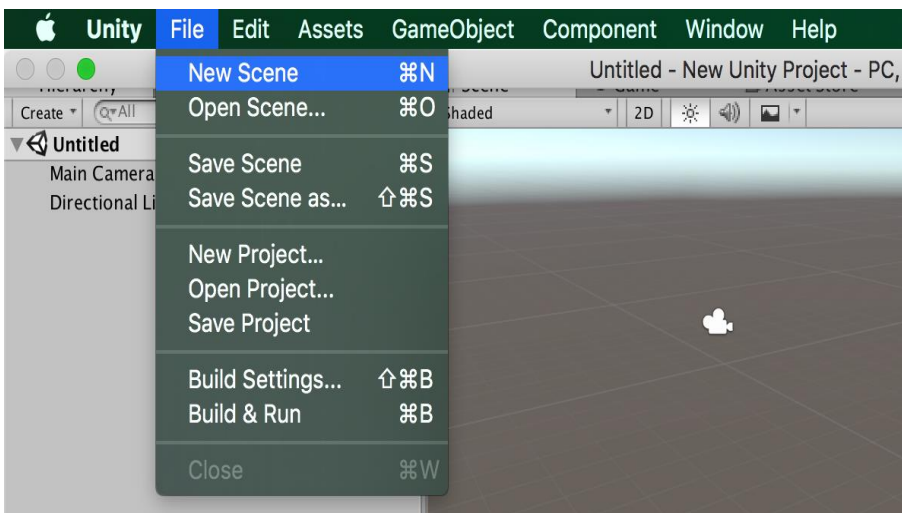
Tabel 5.1 Lingkungan Implementasi Perangkat Lunak

Lingkungan Perangkat Keras	Prosesor : <ul style="list-style-type: none">- Intel(R) Core(TM) i5-4210U CPU @ 2.40GHz Memori : <ul style="list-style-type: none">- 6 GB- VGA Nvidia 820M
Lingkungan Perangkat Lunak	Sistem Operasi : <ul style="list-style-type: none">- Microsoft Windows 10 Home 64-bit Perangkat Pengembang : <ul style="list-style-type: none">- Adobe Photoshop CS5- Unity 5.6.1f1- Visual Studio 2015

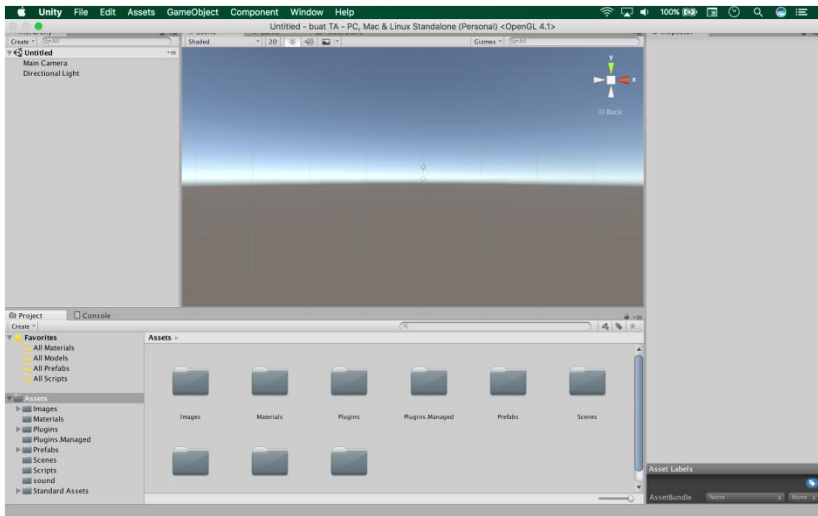
Pada tahap ini implementasi pembuatan aplikasi ini akan dibagi kedalam subbab. Kesepuluh subbab tersebut yaitu meliputi:

4.2 Implementasi Pembuatan Scene

Untuk menambahkan *scene*, hal yang perlu dilakukan adalah menuju tab “*file*” lalu pilih “*new scene*”. Untuk lebih jelasnya, pembuatan scene baru ada di Gambar 4.1. Setelah *scene* terbuat, maka objek-objek yang telah kita miliki terdapat pada *project explorer* yang terletak di kiri bawah aplikasi *Unity* (Gambar 4.3). Untuk memasukkan objek-objek, berkas yang terdapat dalam proyek bisa dimasukkan dengan cara *drag and drop* berkas dari *map* yang terdapat dalam proyek ke *tab scene* (Penjelasan pada Gambar 4.4). Untuk melihat properti objek mengenai posisi atau yang lainnya, pilih objek yang akan dilihat propertinya. Informasi properti terdapat pada label “*Inspector*”. Implementasi dapat dilihat pada Gambar 4.1 sampai Gambar 4.3.



Gambar 5.1 Penambahan scene



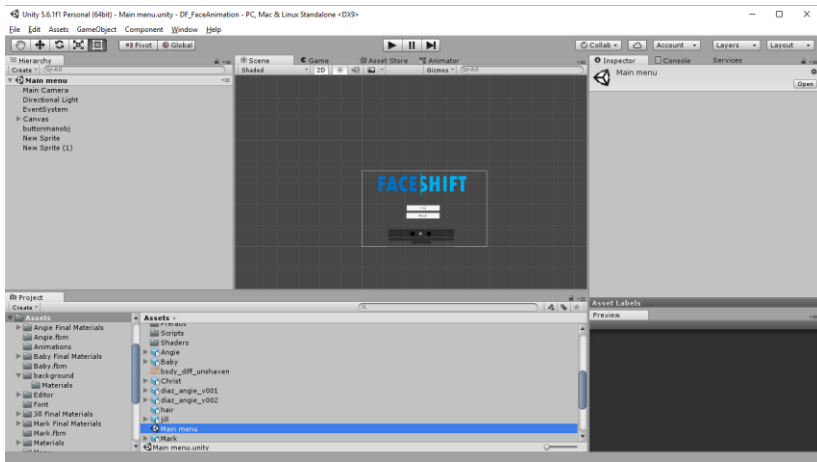
Gambar 5.2 Tampilan scene baru



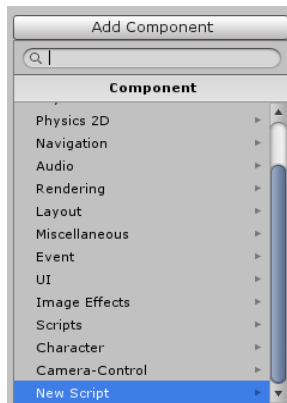
Gambar 5.3 Tampilan menu scene

4.3 Implementasi Pembuatan Script

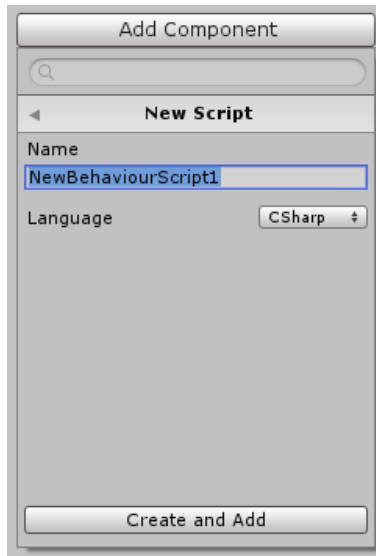
Untuk membuat *script*, dan mengaitkan terhadap objek, pilih objek yang akan dimasukkan *script*, lalu tambahkan komponen script (Gambar 4.4 sampai Gambar 4.6).



Gambar 5.4 Pembuatan Script (1)



Gambar 5.5 Pembuatan Script (2)



Gambar 5.6 Pembuatan *script* (3)

4.4 Implementasi Tampilan Antarmuka Menu

Antarmuka menu yang diimplementasikan oleh penulis antara lain antarmenu awal, antarmuka menu petunjuk. Antarmuka awal merupakan halaman yang ditampilkan ketika pengguna membuka aplikasi. Pada halaman ini terdapat dua tombol yaitu play dan keluar. Tombol mulai akan membuka halaman petunjuk dan keluar untuk menutup aplikasi. Tampilan gambar antarmuka menu dapat dilihat pada Gambar 4.7, Gambar 4.8 dan Gambar 4.9, serta kode implementasi pada Kode 4.1 dan 4.2.



Gambar 5.7 Antarmuka Menu

```

1. public class menumanager : MonoBehaviour {
2.     public void newplayBtn(string newLevel) {
3.         SceneManager.LoadScene(newLevel);
4.     }
5.     public void ExitGameBtn() {
6.         Application.Quit();
7.     }
8.
9. }public class menumanager : MonoBehaviour {
10.     public void newplayBtn(string newLevel) {
11.         SceneManager.LoadScene(newLevel);
12.     }
13.     public void ExitGameBtn() {
14.         Application.Quit();
15.     }
16.
17. }

```

Kode 5.1 tombol main dan keluar

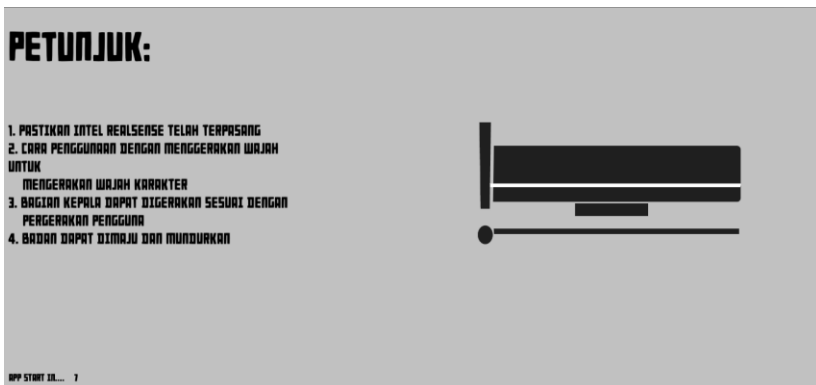
Pada antarmuka apabila ditekan tombol main, akan masuk ke antarmuka petunjuk. Pada antarmuka petunjuk akan menampilkan cara penggunaan dan peringatan untuk memasang kamera *Intel Realsense*. Pada halaman petunjuk memiliki timer selama 10 detik sebelum pindah ke antarmuka utama.


```

18. public class CountTimer : MonoBehaviour {
19.     public string sceneLoad;
20.     private float timer = 10f;
21.     private Text timerSecond;
22.
23.     // Use this for initialization
24.     void Start () {
25.         timerSecond = GetComponent<Text>();
26.     }
27.
28.     // Update is called once per frame
29.     void Update () {
30.         timer -= Time.deltaTime;
31.         timerSecond.text = timer.ToString("f0");
32.         if(timer<=0)
33.         {
34.             Application.LoadLevel(sceneLoad);
35.         }
36.     }
37. }
38.

```

Kode 5.2 Antarmuka petunjuk



Gambar 5.8 Antarmuka petunjuk



Gambar 5.9 Antarmuka utama

4.5 Implementas Mendeteksi Wajah

Agar PC/Laptop dapat mendeteksi kamera Intel Realsense langkah pertama yang harus dilakukan adalah menginstal DCM (Deep Camera Manager) Intel Realsense Driver pada <https://software.intel.com/en-us/intel-realsense-sdk/download>. Kemudian langkah selanjutnya adalah menginstall Intel Realsense SDK pada <https://software.intel.com/en-us/intel-realsense-sdk/download>.

Agar kamera dapat mengambil landmark data pada pengguna. Sebuah variable *isposevalid* pengecek wajah diset false. Kemudian wajah yang terekam dicek semua wajah yang telah didapat pada kamera pertama kali dimasukan ke dalam sebuah array face. Apabila faceindex lebih dari 0 maka data wajah telah didapat dan diambil wajah terdekat dengan kamera. Kemudian data dari faceindex dicek apakah memiliki semua data yang diinginkan atau tidak, contohnya dicek apakah gambar wajah dapat dipasang semua landmark poin atau tidak. Apabila tidak maka akan muncul debug

bahwa data landmark error. Setelah semua data ada maka variable *isposevalid* diset true. Kode validasi dapat dilihat pada Kode 4.3.

```

39. private void UpdateFace()
40. {
41.     _pxcmFaceData.Update();
42.     _isPoseValid = false;
43.     // Get the closest face
44.     PXCMFaceData.Face pxcmFace = _pxcmFaceData.QueryFaceByIndex(0);
45.     if (_pxcmFaceData.QueryNumberOfDetectedFaces() <= 0) {
46.         Debug.Log("Gag dapet wajah");
47.         return; }
48.     else if (_pxcmFaceData.QueryNumberOfDetectedFaces() > 0) {
49.         Debug.Log("dapet WAJAH");
50.         pxcmStatus RealSenseCameraStatus = _pxcmSenseManager.AcquireFrame(false, 0);
51.         switch(RealSenseCameraStatus) {
52.             case pxcmStatus.PXCM_STATUS_NO_ERROR:
53.                 PXCMEmotion ft = _pxcmSenseManager.QueryEmotion();
54.                 if (ft != null){
55.                     ProcessEmotion(ft); }
56.                 _pxcmSenseManager.ReleaseFrame();
57.                 break; } }
58.     PXCMFaceData.LandmarksData pxcmLandmarksData = pxcmFace.QueryLandmarks();
59.     if (pxcmLandmarksData == null) {
60.         Debug.Log("PXCMFaceData.Face.QueryLandmarks() failed.");
61.         return; }
62.     PXCMFaceData.PoseData pxcmPoseData = pxcmFace.QueryPose();
63.     if (pxcmPoseData == null) {
64.         Debug.Log("PXCMFaceData.Face.QueryPose() failed.");
65.         return; }
66.     PXCMFaceData.LandmarkPoint[] pxcmLandmarkPoints = new PXCMFaceData.LandmarkPoint[_maxLandmarks];

```

```

67.         for (int i = 0; i < _maxLandmarks; ++i){
68.             pxcmLandmarkPoints[i] = new PXCMFaceData.
                LandmarkPoint(); }
69.         bool result = pxcmLandmarksData.QueryPoints(o
ut pxcmLandmarkPoints);
70.         if (!result) {
71.             Debug.Log("PXCMFaceData.LandmarksData.Que
ryPoints() failed.");
72.             return; }
73.         PXCMFaceData.ExpressionsData pxcmExpressionsD
ata = pxcmFace.QueryExpressions();
74.         if (pxcmExpressionsData == null) {
75.             Debug.Log("PXCMFaceData.Face.QueryExpress
ions() failed.");
76.             return; }
77.         _isPoseValid = true;

```

Kode 5.3 kode validasi data wajah

Kemudian diambil distance dari masing masing landmark untuk didapatkan data dari wajah pengguna. Kode ambil data pada wajah dapat dilihat pada Kode 4.4.

```

78. NEUTRAL_LBROW_UP_DISTANCE = GetLandmarksDistance (pxc
mLandmarkPoints [7].image.y, pxcmLandmarkPoints [77].
image.y);
79.         NEUTRAL_RBROW_UP_DISTANCE = GetLandmarksD
istance (pxcmLandmarkPoints [2].image.y, pxcmLandmark
Points [76].image.y);

```

Kode 5.4 ambil data distance pada wajah

Setelah itu disetting apakah mode kamera dalam mode mirror atau tidak. Kode mirror mode dapat dilihat pada Kode 4.5.

```

1. if (_cameraMirrorMode == PXCMCapture.Device.MirrorMod
e.MIRROR_MODE_HORIZONTAL) {
2.         _lBrowUp = RBrowUp;
3.         _rBrowUp = LBrowUp;
4.         _lBlink = RBlink;

```

```

5.         _rBlink = LBlink;
6.         _jawMoveLeftRight = -
    jawMoveLeftRight;    }
7.     else {
8.         _lBrowUp = LBrowUp;
9.         _rBrowUp = RBrowUp;
10.        _lBlink = LBlink;
11.        _rBlink = RBlink;
12.        _jawMoveLeftRight = jawMoveLeftRight;    }

1.  /// Update head
2.      pxcmPoseEulerAngles.pitch *= -1;
3.      pxcmPoseEulerAngles.roll *= -1;
4.      if (_cameraMirrorMode == PXCMAbstractDevice.M
    irrorMode.MIRROR_MODE_HORIZONTAL) {
5.          currentHeadPose = new Vector3((float)pxcm
    PoseEulerAngles.yaw, -
    (float)pxcmPoseEulerAngles.pitch, (float)pxcmPoseEule
    rAngles.roll);    }
6.      else {
80.         currentHeadPose = new Vector3((float)pxcm
    PoseEulerAngles.yaw, -
    (float)pxcmPoseEulerAngles.pitch, (float)pxcmPoseEule
    rAngles.roll);    }

```

Kode 5.5 kode mirror pada kamera

Data landmark per bagian wajah kemudian disimpan ke dalam data float yang nanti digunakan di face animation, Kode menyimpan ekspresi pada wajah, kepala dan badan serta menampilkan landmark pada gambar pengguna dapat dilihat pada Kode 4.6 sampai 4.9.

```

81. _rBrowUp = GetNormalizedLandmarksDistance(GetLandmark
    sDistance(pxcmlandmarkPoints[2].image.y, pxcmlandmark
    Points[76].image.y, pxcmPoseEulerAngles.pitch), NEUTRA
    L_RELATIVE_FACE_HEIGHT, faceHeight, NEUTRAL_RBROW_UP_
    DISTANCE, TARGET_RBROW_UP_DISTANCE, NORMALIZE_MAX_FAC
    IAL_EXPRESSION_VALUE, true);
82. _lBrowDown = GetNormalizedLandmarksDistance(GetLandma
    rksDistance(pxcmlandmarkPoints[7].image.y, pxcmlandma
    rkPoints[77].image.y, pxcmPoseEulerAngles.pitch), NEU

```

```

    TRAL_RELATIVE_FACE_HEIGHT, faceHeight, NEUTRAL_LBROW_
    DOWN_DISTANCE, TARGET_LBROW_DOWN_DISTANCE, NORMALIZE_
    MAX_FACIAL_EXPRESSION_VALUE, true);

```

Kode 5.6 kode menyimpan ekspresi wajah

```

83.         // Smooth the raw headpose data
84.         currentHeadPose = Vector3.Lerp(_oldHeadPose,
    currentHeadPose, SMOOTH_FACTOR);
85.         _oldHeadPose = currentHeadPose;
86.         // Account for calibration offset
87.         currentHeadPose -= _neutralHeadPose;
88.         _headYaw = currentHeadPose.x;
89.         _headPitch = currentHeadPose.y;
90.         _headRoll = currentHeadPose.z
91.

```

Kode 5.7 menyimpan pergerakan kepala

```

92. // Update back
93.     _pxcmFaceDetectionData = pxcmFace.QueryDetect
    ion();
94.     float faceAverageDepth = 0f;
95.     result = _pxcmFaceDetectionData.QueryFaceAver
    ageDepth(out faceAverageDepth);
96.     if (result == true) {
97.         if (_baseFaceAverageDepth == -1) {
98.             _baseFaceAverageDepth = faceAverageDe
    pth;
99.             _backPitch = 0f; }
100.        else {
101.            _backPitch = Mathf.Lerp(_back
    Pitch, (faceAverageDepth - _baseFaceAverageDepth) / M
    AX_BACK_PITCH * NORMALIZE_MAX_BACK_PITCH_VALUE, SMOOT
    H_FACTOR); } }
102.        else {
103.            _baseFaceAverageDepth = -1;
104.            _backPitch = 0f;
105.            Debug.Log("PXCmFaceData.Face.Query
    Detection() failed."); } }
106.        if (!_isPoseInitialized) {

```

```

107.             _isPoseInitialized = true;
108.             StartCalibration();    }
109.     private void StartCalibration() {
110.         if (_isCalibrating) {
111.             return; }

```

Kode 5.8 Kode menyimpan pergerakan badan

```

112.     private void DrawLandmarks(PXCMFaceData.Landm
arkPoint[] pxcmlandmarkPoints) {
113.         // Draw all landmarks
114.         Color32[] pixels = _operatorVideoText
ure.GetPixels32();
115.         for (int i = 0; i < pxcmlandmarkPoint
s.Length; i++) {
116.             // Make it 4 pixels big
117.             int x = Mathf.Min(Mathf.Max((int)pxcml
andmarkPoints[i].image.x, 0), _operatorVideoTexture.w
idth);
118.             int y = Mathf.Min(Mathf.Max(_operatorV
ideoTexture.height -
(int)pxcmlandmarkPoints[i].image.y, 0), _operatorVide
oTexture.height);
119.             Color32 color = new Color32(0, 0, 0, 1
00);;
120.             if (x < _operatorVideoTexture.width - 2
&& y < _operatorVideoTexture.height - 2) {
121.                 pixels[(y + 2) * _operatorVid
eoTexture.width + x] = color;
122.                 pixels[(y + 2) * _operatorVid
eoTexture.width + x + 1] = color;
123.                 pixels[(y) * _operatorVideoTe
xture.width + x + 2] = color;
124.                 pixels[(y + 1) * _operatorVid
eoTexture.width + x + 2] = color;
125.                 pixels[(y + 2) * _operatorVid
eoTexture.width + x + 2] = color;
126.
127.                 pixels[y * _operatorVideoText
ure.width + x] = color;

```

```

128.         pixels[y * _operatorVideoText
           ure.width + x + 1] = color;
129.         pixels[(y + 1) * _operatorVid
           eoTexture.width + x] = color;
130.         pixels[(y + 1) * _operatorVid
           eoTexture.width + x + 1] =color;           }           }
131.
132.         _operatorVideoTexture.SetPixels32(pixels);
133.         _operatorVideoTexture.Apply();

```

Kode 5.9 Kode menampilkan landmark pada gambar wajah pengguna



Gambar 5.10 Antarmuka utama

4.6 Implementasi pendeteksi Emosi Pengguna

Saat pengguna menggerakkan wajah, secara tidak langsung pengguna menampilkan emosi dari pengguna itu sendiri. *Intel* dapat membedakan bermacam ekspresi dari pengguna. Caranya adalah:

6. Membuat list sentiment dan emotion.
 7. Memasukan secara manual data dari sentiment dan emotion.
 8. Menggambil emotion dari query emotion
 9. High evidence diset -10 kemudian dicari dari setiap emotion data yang mana memiliki evidence paling tinggi
 10. Apabila ada 2 emotion atau sentiment memiliki evidence yang sama maka dicari intensity paling tinggi
 11. Ditampilkan pada text emotion dan sentiment pada monitor
- Gambar implementasi dan kode dapat dilihat pada Gambar 4.11 dan 4.12 dan Kode 4.10 sampai 4.12.

Main Face			
▼ Sentiments			
Size	3		
▼ Positive			
Name	Positive		
Emotion	EMOTION_SENTIMENT_POSITIVE ↑		
Intensity	0		
Evidence	0		
▼ Neutral			
Name	Neutral		
Emotion	EMOTION_SENTIMENT_NEUTRAL ↑		
Intensity	0		
Evidence	0		
▼ Negative			
Name	Negative		
Emotion	EMOTION_SENTIMENT_NEGATIVE ↑		
Intensity	0		
Evidence	0		
▼ Emotions			
Size	7		
▼ HAPPY			
Name	HAPPY		
Emotion	EMOTION_PRIMARY_JOY ↑		
Intensity	0		
Evidence	0		
▼ ANGER			
Name	ANGER		
Emotion	EMOTION_PRIMARY_ANGER ↑		
Intensity	0		
Evidence	0		
▼ SAD			
Name	SAD		
Emotion	EMOTION_PRIMARY_SADNESS ↑		
Intensity	0		

Gambar 5.11 Memasukan data emotion dan sentiment

```

134.     _pxcmFaceData.Update();
135.     _isPoseValid = false;
136.         // Get the closest face
137.         PXCMFaceData.Face pxcmFace = _pxcmFaceData.QueryFaceByIndex(0);
138.         if (_pxcmFaceData.QueryNumberOfDetectedFaces() <= 0) { Debug.Log("Gag dapet wajah");

139.             return; }
140.         else if (_pxcmFaceData.QueryNumberOfDetectedFaces() > 0) {
141.             Debug.Log("dapet WAJAH");
142.             pxcmStatus RealSenseCameraStatus
= _pxcmSenseManager.AcquireFrame(false, 0);
143.             switch(RealSenseCameraStatus) {

144.                 case pxcmStatus.PXCM_STATUS_NO_ERROR:
145.                     PXCMEmotion ft = _pxcmSenseManager.QueryEmotion();
146.                     //Debug.Log(ft);
147.                     if (ft != null) {
148.                         ProcessEmotion(ft);
149.                     }
150.                     _pxcmSenseManager.ReleaseFrame();

                     break; } }

```

Kode 5.10 Kode mendapatkan emotion dari query emotion

```

1. public class EmotionTrackedFace {
2.     public List<EmotionStatus> Sentiments;
3.     public List<EmotionStatus> Emotions;
4.     public EmotionStatus FindMostLikely(List<EmotionStatus> emotions) {
5.         int HighestEvidence = -10;
6.         Debug.Log("emotion count " + emotions.Count);
7.         for (int i = 0; i < emotions.Count; i++)
            {
                if (emotions[i].evidence > HighestEvidence)
                    HighestEvidence = emotions[i].evidence;
            }
    }
}

```

```

8.         List<EmotionStatus> Likely = new List<Emotion
9.           Status>();
10.        for (int i = 0; i < emotions.Count; i++) {
11.            if (emotions[i].evidence == HighestEvide
12.              nce) {
13.                Likely.Add(emotions[i]);      }
14.                EmotionStatus MostLikely = Likely[0];
15.                for (int i = 0; i < Likely.Count; i++) {
16.                    if (Likely[i].intensity > MostLikely.int
17.                      ensity)
18.                        MostLikely = Likely[i];      }
19.        }
20.        return MostLikely; }

```

Kode 5.11 Kode mendapatkan sentiment dan emotion tertinggi

```

17. public void ProcessEmotion(PXCMEemotion emotions)
18. {
19.     NumberOfFaces = emotions.QueryNumFaces();
20.     //Debug.Log(NumberOfFaces);
21.     if (NumberOfFaces > 0)
22.     {
23.         foreach (var emo in MainFace.Emotions)
24.         {
25.             //Debug.Log("Main Face Emotion " + Mai
26.               nFace.Emotions);
27.             PXCMEemotion.EmotionData emotiondata;
28.             emotions.QueryEmotionData(0, emo.emoti
29.               on, out emotiondata);
30.             emo.intensity = emotiondata.intensity;
31.             emo.evidence = emotiondata.evidence;
32.         }
33.         foreach (var emo in MainFace.Sentiments)
34.         {
35.             PXCMEemotion.EmotionData emotiondata;
36.             emotions.QueryEmotionData(0, emo.emoti
37.               on, out emotiondata);
38.             emo.intensity = emotiondata.intensity;

```

```

37.             emo.evidence = emotiondata.evidence;
38.         }
39.
40.         CurrentEmotion = MainFace.FindMostLikely(M
ainFace.Emotions);
41.         GeneralSentiment = MainFace.FindMostLikely
(MainFace.Sentiments);
42.     }
43.     textemosi.text = "" + CurrentEmotion.name;
44.     textsentiment.text = "" + GeneralSentiment.nam
e;

```

Kode 5.12 Kode menyimpan data emotion dan sentiment



Gambar 5.12 Tampilan emosi pada aplikasi

4.7 Implementasi Menggerakkan Animasi Karakter

Intel Realsense SDK memiliki fitur *face animation* yaitu fitur untuk menggerakkan wajah dan badan karakter dengan input dari wajah pengguna.

Berikut adalah kode untuk mendeteksi karakter, menggunakan data wajah pengguna hingga menggerakkan karakter. Implementasi gambar dan kode dapat dilihat pada Gambar 4.13 sampai 4.15 dan Kode 4.13 sampai dengan 4.18.

```

45. public int _rBrowDownBodyBlendShapeIndex = -1;
46. public int _lBrowDownBodyBlendShapeIndex = -1;
47. public int _rBrowUpBodyBlendShapeIndex = -1;
48. public int _lBrowUpBodyBlendShapeIndex = -1;
49. public int _rBrowInBodyBlendShapeIndex = -1;
50. public int _lBrowInBodyBlendShapeIndex = -1;
51. public int _lBlinkBodyBlendShapeIndex = -1;
52. public int _rBlinkBodyBlendShapeIndex = -1;
53. public int _lLowerLidUpBodyBlendShapeIndex = -1;
54. public int _rLowerLidUpBodyBlendShapeIndex = -1;
55. public int _lSmileBodyBlendShapeIndex = -1;
56. public int _rSmileBodyBlendShapeIndex = -1;
57. public int _lSmileCorrectiveBodyBlendShapeIndex = -
    1;
58. public int _rSmileCorrectiveBodyBlendShapeIndex = -
    1;
59. public int _lMouthSqueezeBodyBlendShapeIndex = -1;
60. public int _rMouthSqueezeBodyBlendShapeIndex = -1;
61. public int _upperLipPressBodyBlendShapeIndex = -1;
62. public int _lowerLipPressBodyBlendShapeIndex = -1;
63. public int _upperLipUpBodyBlendShapeIndex = -1;
64. public int _lowerLipDownBodyBlendShapeIndex = -1;
65. public int _mouthFrownBodyBlendShapeIndex = -1;
66. public int _mouthPuckerBodyBlendShapeIndex = -1;
67. public int _mouthOpenBodyBlendShapeIndex = -1;
68. public int _mouthOpenMouthBlendShapeIndex = -1;
69. //EyeLens BlendShapeIndex
70. public int _lensRightEyeSmileBlendShapeIndex = -1;
71. public int _lensRightEyeBlinkBlendShapeIndex = -1;
72. public int _lensRightEyeOpenBlendShapeIndex = -1;
73. public int _lensRightEyeLookUpBlendShapeIndex = -1;
74. public int _lensLeftEyeSmileBlendShapeIndex = -1;
75. public int _lensLeftEyeBlinkBlendShapeIndex = -1;
76. public int _lensLeftEyeOpenBlendShapeIndex = -1;
77. public int _lensLeftEyeLookUpBlendShapeIndex = -1;

```

Kode 5.13 Kode Insiasi value bagian dari objek karakter

R Brow Down Body	0
L Brow Down Body	1
R Brow Up Body Blend	2
L Brow Up Body Blend	3
R Brow In Body Blend	5
L Brow In Body Blend	4
L Blink Body Blend	6
R Blink Body Blend	7
L Lower Lid Up Body	8
R Lower Lid Up Body	9
L Smile Body Blend	10
R Smile Body Blend	11
L Smile Corrective	12
R Smile Corrective	13
L Mouth Squeeze Body	-1
R Mouth Squeeze Body	-1
Upper Lip Press Body	16
Lower Lip Press Body	17
Upper Lip Up Body	19
Lower Lip Down Body	18
Mouth Frown Body	20
Mouth Pucker Body	21
Mouth Open Body Blend	22
Mouth Open Mouth	0
Lens Right Eye Smile	-1
Lens Right Eye Blink	-1
Lens Right Eye Open	-1
Lens Right Eye Look	-1
Lens Left Eye Smile	-1
Lens Left Eye Blink	-1
Lens Left Eye Open	-1
Lens Left Eye Look	-1

Gambar 5.13 Tampilan nilai lokasi objek pada karakter

```

78. private float _lBrowDownInitialValue = 0f;
79.     private float _rBrowDownInitialValue = 0f;
80.     private float _lBrowUpInitialValue = 0f;
81.     private float _rBrowUpInitialValue = 0f;
82.     private float _lBrowInInitialValue = 0f;
83.     private float _rBrowInInitialValue = 0f;
84.     private float _lBlinkInitialValue = 0f;

```

```

85.     private float _rBlinkInitialValue = 0f;
86.     private float _lLowerLidUpInitialValue = 0f;
87.     private float _rLowerLidUpInitialValue = 0f;
88.     private float _lSmileInitialValue = 0f;
89.     private float _rSmileInitialValue = 0f;
90.     private float _lSmileCorrectiveInitialValue = 0f
;
91.     private float _rSmileCorrectiveInitialValue = 0f
;
92.     private float _lMouthSqueezeInitialValue = 0f;
93.     private float _rMouthSqueezeInitialValue = 0f;
94.     private float _upperLipPressInitialValue = 0f;
95.     private float _lowerLipPressInitialValue = 0f;
96.     private float _lowerLipDownInitialValue = 0f;
97.     private float _upperLipUpInitialValue = 0f;
98.     private float _mouthFrownInitialValue = 0f;
99.     private float _mouthPuckerInitialValue = 0f;
100.    private float _mouthOpenInitialValue = 0f
;
101.    private float _mouthOpenMouthInitialValue
= 0f;
102.
103.
104.    //EyeLensInitials
105.    private float _lensRightEyeSmileInitialVa
lue = 0f;
106.    private float _lensRightEyeBlinkInitialVa
lue = 0f;
107.    private float _lensRightEyeOpenInitialVal
ue = 0f;
108.    private float _lensRightEyeLookUpInitialV
alue = 0f;
109.    private float _lensLeftEyeSmileInitialValue =
0f;
110.    private float _lensLeftEyeBlinkInitialVal
ue = 0f;
111.    private float _lensLeftEyeOpenInitialValu
e = 0f;
112.    private float _lensLeftEyeLookUpInitialVa
lue = 0f;

```

Kode 5.14 Kode inisiasi indeks pergerakan objek animasi

Kemudian setiap object dicek apakah di dalam karakter memiliki indeks, apabila lebih dari 0 maka nilai yang ada di dalam indeks dipindahkan kedalam value sesuai dengan variabelnya.

```

113.         _bodyRenderer = bodyObject? _bodyObject.GetCo
mponent<SkinnedMeshRenderer>() : null;
114.         _mouthRenderer = mouthObject? _mouthO
bject.GetComponent<SkinnedMeshRenderer>() : null;
115.         _lensEyesRenderer = _lensNiceObject?
_mouthObject.GetComponent<SkinnedMeshRenderer>() : n
ull;
116.         if (_bodyRenderer) {#region BodyChe
ckIndex
117.             if (_rBrowDownBodyBlendShapeIndex
>= 0) {
118.                 _rBrowDownInitialValue = _bod
yRenderer.GetBlendShapeWeight (_rBrowDownBodyBlendSh
apeIndex); }             if (_lBrowDownBodyBlendShap
eIndex >= 0) {
119.                 _lBrowDownInitialValue = _bod
yRenderer.GetBlendShapeWeight (_lBrowDownBodyBlendSh
apeIndex); }             if (_rBrowUpBodyBlendShapeI
ndex >= 0) {
120.                 _rBrowUpInitialValue = _bodyR
enderer.GetBlendShapeWeight (_rBrowUpBodyBlendShapeI
ndex); }
121.             if (_lBrowUpBodyBlendShapeIndex >
= 0) {
122.                 _lBrowUpInitialValue = _bodyR
enderer.GetBlendShapeWeight (_lBrowUpBodyBlendShapeI
ndex);

```

Kode 5.15 Kode Sumber Fungsi untuk mengecek karakter

```

123.         float rBrowDown = 0;
124.         float lBrowDown = 0;
125.         float rBrowUp = 0;
126.         float lBrowUp = 0;
127.         float rBrowIn = 0;

```



```

128.         float lBrowIn = 0;
129.         float rBlink = 0;
130.         float lBlink = 0;
131.         float rLowerLidUp = 0;
132.         float lLowerLidUp = 0;
133.         float rSmile = 0;
134.         float lSmile = 0;
135.         float rSmileCorrective = 0;
136.         float lSmileCorrective = 0;
137.         float lMouthSqueeze = 0;
138.         float rMouthSqueeze = 0;
139.         float upperLipPress = 0;
140.         float lowerLipPress = 0;
141.         float lowerLipDown = 0;
142.         float upperLipUp = 0;
143.         float mouthFrown = 0;
144.         float mouthPucker = 0;
145.         float mouthOpen = 0;
146.         float mouthOpenMouth = 0;
147.         //EyeLensValues
148.         float lensRightEyeSmile = 0;
149.         float lensRightEyeBlink = 0;
150.         float lensRightEyeOpen = 0;
151.         float lensRightEyeLookUp = 0;
152.         float lensLeftEyeSmile = 0;
153.         float lensLeftEyeBlinkx = 0;
154.         float lensLeftEyeOpen = 0;
155.         float lensLeftEyeLookUp = 0;

```

Kode 5.16 Kode Sumber nilai dari facetracking

```

1.  if (_faceTrackingManager && !_faceTrackingManager._m
    anualMode)    {
1.      rBrowDown = _faceTrackingManager.RBrowDown;
    wn;          lBrowDown = _faceTrackingManager.LBrowDown;
    wDown;       rBrowUp = _faceTrackingManager.RBrowUp;
    owUp;
2.      lBrowUp = _faceTrackingManager.LBrowUp;
3.      rBrowIn = _faceTrackingManager.RBrowIn;

```

```

4.          lBrowIn = _faceTrackingManager.LBrowIn;
5.          rBlink = _faceTrackingManager.RBlink;
6.          lBlink = _faceTrackingManager.LBlink;
7.          lLowerLidUp = _faceTrackingManager.LLowL
   idUp;
8.          rLowerLidUp = _faceTrackingManager.RLowL
   idUp;
9.          lSmile = _faceTrackingManager.LMouthSmil
   e;          rSmile = _faceTrackingManager.RMouthS
   mile;          lSmileCorrective = _faceTrackingMa
   nager.LMouthSmileCorrect;
10.         rSmileCorrective = _faceTrackingManager.
   RMouthSmileCorrect;
11.         lMouthSqueeze = _faceTrackingManager.LMo
   uthSqueeze;
12.         rMouthSqueeze = _faceTrackingManager.RMo
   uthSqueeze;
13.         upperLipPress = _faceTrackingManager.Upp
   erLipPress;
14.         lowerLipPress = _faceTrackingManager.Low
   erLipPress;
15.         lowerLipDown = _faceTrackingManager.Lowe
   rLipDown;
16.         upperLipUp = _faceTrackingManager.UpperL
   ipUp;
17.         mouthFrown = _faceTrackingManager.MouthF
   rown;
18.         mouthPucker = _faceTrackingManager.Mouth
   Pucker;
19.         mouthOpen = mouthOpenMouth = _faceTracki
   ngManager.MouthOpen;
20.         lensRightEyeSmile = 0;
21.         lensRightEyeBlink = 0;
22.         lensRightEyeOpen = 0;
23.         lensRightEyeLookUp = 0;
24.         lensLeftEyeSmile = 0;
25.         lensLeftEyeBlinkx = 0;
26.         lensLeftEyeOpen = 0;
156.         lensLeftEyeLookUp = 0;

```

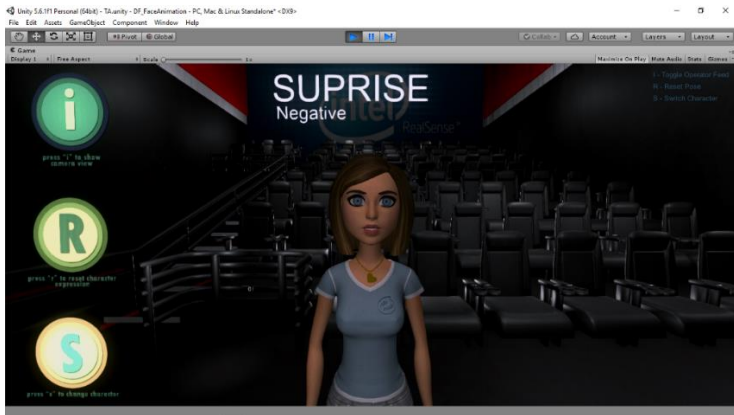
Kode 5.17 Kode Sumber value yang digunakan untuk menerima data facetracking kedalam data indeks karakter

```

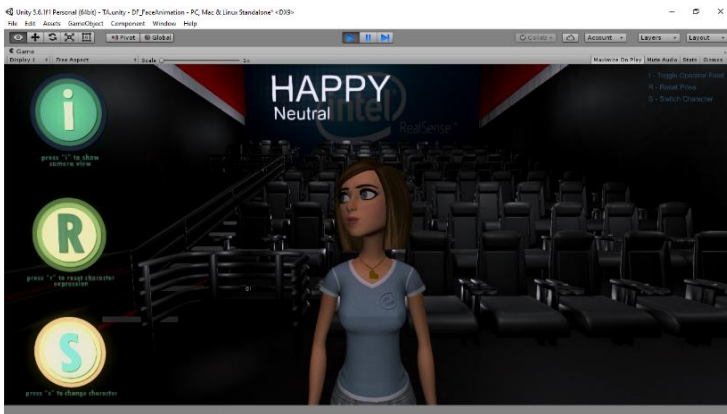
157.         float value;
158.         //BodyRenderer
159.         SetBlendShapeValue (rBrowUp, _rBrowUp
BodyBlendShapeIndex, _rBrowUpInitialValue, ref _exec
uteRBrowUpSmoothing, out value, _bodyRenderer);
160.         SetBlendShapeValue (lBrowUp, _lBrowUp
BodyBlendShapeIndex, _lBrowUpInitialValue, ref _exec
uteLBrowUpSmoothing, out value, _bodyRenderer);
161.         SetBlendShapeValue (rBrowDown, _rBrow
DownBodyBlendShapeIndex, _rBrowDownInitialValue, ref
_executeRBrowDownSmoothing, out value, _bodyRenderer
);
162.         SetBlendShapeValue (lBrowDown, _lBrow
DownBodyBlendShapeIndex, _lBrowDownInitialValue, ref
_executeLBrowDownSmoothing, out value, _bodyRenderer
);
163.         SetBlendShapeValue (rBrowIn, _rBrowIn
BodyBlendShapeIndex, _rBrowInInitialValue, ref _exec
uteRBrowInSmoothing, out value, _bodyRenderer);
164.         SetBlendShapeValue (lBrowIn, _lBrowIn
BodyBlendShapeIndex, _lBrowInInitialValue, ref _exec
uteLBrowInSmoothing, out value, _bodyRenderer);
165.         SetBlendShapeValue (rBlink, _rBlinkBo
dyBlendShapeIndex, _rBlinkInitialValue, ref _execute
RBlinkSmoothing, out value, _bodyRenderer);
166.         SetBlendShapeValue (lBlink, _lBlinkBo
dyBlendShapeIndex, _lBlinkInitialValue, ref _execute
LBlinkSmoothing, out value, _bodyRenderer);
167.         SetBlendShapeValue (rLowerLidUp, _rLo
werLidUpBodyBlendShapeIndex, _rLowerLidUpInitialValu
e, ref _executeRLowerLidUpSmoothing, out value, _bod
yRenderer);
168.         SetBlendShapeValue (lLowerLidUp, _lLo
werLidUpBodyBlendShapeIndex, _lLowerLidUpInitialValu
e, ref _executeLLowerLidUpSmoothing, out value, _bod
yRenderer);
169.         SetBlendShapeValue (rSmile, _rSmileBo
dyBlendShapeIndex, _rSmileInitialValue, ref _execute
RSmileSmoothing, out value, _bodyRenderer);

```

Kode 5.18 Kode Sumber untuk menggerakan karakter dari karakter sesuai data dari indeks karakter



Gambar 5.14Tampilan Karakter sebelum bergerak



Gambar 5.15 Tampilan Karakter sesudah bergerak

4.8 Implementasi Ganti Karakter

Pengguna dapat mengganti karakter sesuai dengan jumlah karakter yang telah disediakan oleh aplikasi. Pertama dibuat list

karakter, dan karakter pertama dijadikan current character. Pertama current character indeks diset 0 dan previouscharacter indeks diset -1. Ketika pengguna mengaktifkan ganti karakter, pertama kali indeks current character dimasukan ke dalam nilai indeks previous karakter. Sedangkan nilai current character +1 kemudian dimodulus dengan jumlah karakter yang ada. Sehingga current character sekarang menjadi 1, maka diambil karakter dan environment dengan indeks yang sama, kemudian dipindahkan ke lokasi x, y dan z yang diinginkan dan kemudian setActive di set true. Begitu juga sebaliknya previous character indeks dengan setActive di set false. Implementasi ganti karakter dapat dilihat pada Gambar 4.16 dan 4.17, dan pada Kode 4.19.

```

151.     private FaceTrackingManager _faceTrackingMana
    ger = null;
152.     private int _previousCharacterIndex = -
    1;
153.     private int _currentCharacterIndex = 0;
154.
155.     if (Event.current.type == EventType.KeyDown)
    {
156.         if (Event.current.keyCode == KeyC
    ode.S) {
157.             _previousCharacterIndex = _cu
    rrentCharacterIndex;
158.             _currentCharacterIndex = (_cu
    rrentCharacterIndex + 1) % _characterObjects.Length;
159.
160.             SetCurrentCharacter(); }
    }
161.     private void SetCurrentCharacter() {
162.         if (_faceTrackingManager) {
163.             if ((_currentCharacterIndex >= 0)
    && (_currentCharacterIndex < _characterObjects.Length
    )) {
164.                 if (_characterObjects[_curren
    tCharacterIndex]) {

```

```

165.         Transform currentEnvironmentTransf
orm = _characterObjects[_currentCharacterIndex].trans
form.Find("Environment");
166.         if (currentEnvironmentTra
nsform){
167.             currentEnvironmentTra
nsform.gameObject.SetActive(true);        }
168.             GameObject currentCharacte
r = _characterObjects[_currentCharacterIndex];
169.             Vector3 position = new Ve
ctor3(0, 0, 0);
170.             currentCharacter.transform.posit
ion = new Vector3(position.x, currentCharacter.transf
orm.position.y, position.z);
171.         } }
172.         if ((_previousCharacterIndex >=
0) && (_previousCharacterIndex < _characterObjects.Le
ngth)) {
173.             if (_characterObjects[_previo
usCharacterIndex]) {
174.                 Transform previousEnvironment
Transform = _characterObjects[_previousCharacterIndex
].transform.Find("Environment");
175.                 if (previousEnvironmentTransf
orm) {
176.                     previousEnvironmentTransform.gameObject.SetAc
tive(false);    }
177.                     GameObject previousCharacter =
_characterObjects[_previousCharacterIndex];
178.                     Vector3 position = new Vector3(
100, 100, 100);
179.                     previousCharacter.transform.pos
ition = new Vector3(position.x, previousCharacter.tran
sform.position.y, position.z);}    } }
180.

```

Kode 5.19 kode ganti karakter



Gambar 5.16 Tampilan Karakter sebelum diganti



Gambar 5.17 Tampilan Karakter sesudah diganti

(Halaman ini sengaja dikosongkan)

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas hal mengenai uji coba dan evaluasi aplikasi manasik realitas virtual ibadah haji dengan menggunakan Oculus Rift. Uji coba dilakukan menggunakan metode black box (kotak hitam) berdasarkan skenario yang telah ditentukan. Uji coba dilakukan pada 2 buah komputer terhadap hasil implementasi perangkat lunak yang telah dijelaskan pada bab IV.

5.1 Lingkungan Uji Coba

Pada proses uji coba ini, lingkungan dibedakan menjadi lingkungan perangkat keras dan perangkat lunak. Berikut ini akan dijelaskan mengenai tiap-tiap lingkungan uji coba aplikasi.

5.2 Lingkungan Perangkat Keras

Lingkungan pelaksanaan uji coba meliputi perangkat keras dan perangkat lunak yang akan digunakan pada sistem ini. Spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam rangka uji coba perangkat lunak ini dicantumkan pada Tabel 5.1:

Tabel 6.1 Tabel perangkat keras

Deskripsi
Prosesor : -Intel(R) Core(TM) i5-4210U CPU @ 2.40GHz
Memori : 6.00 GB

5.3. Lingkungan Perangkat Lunak

Deskripsi perangkat lunak untuk proses uji coba dapat dilihat pada Tabel 5.2.

Tabel 6.2 Tabel perangkat lunak

	Deskripsi
	Sistem Operasi Windows 10 Home 64-bit.
	Oculus Runtime SDK.

5.4 Pengujian Fungsionalitas

Untuk mengetahui kesesuaian keluaran dari tiap tahap dan langkah penggunaan fitur terhadap skenario yang dipersiapkan, maka dibutuhkan pengujian fungsionalitas. Penjabaran skenario dan hasil uji coba fungsionalitas yang dilakukan terhadap perangkat lunak yang dibangun akan dijabarkan pada subbab ini.

5.4.1 Skenario Pengujian Mendeteksi Wajah

Tabel 6.3 Ujicoba UC-001

Nama Kasus Penggunaan	Mendeteksi wajah pengguna
Kode	UC-001
Kondisi awal	Pemain telah memasuki menu utama
Proses pengujian	Pengguna menggerakkan kepala dan wajah
Hasil yang diharapkan	Terdapat landmark data pada foto pengguna

Nama Kasus Penggunaan	Mendeteksi wajah pengguna
Hasil yang diperoleh	Landmark berhasil ditambahkan
Kesimpulan	Berhasil



Gambar 6.1 Hasil pengujian mendeteksi wajah pengguna

5.4.2 Skenario Pengujian Menggerakan Kepala

1. Pengujian Kepala secara *Yaw*

Tabel 6.4 Tabel Ujicoba UC-002

Nama Kasus Penggunaan	Mengerakan kepala secara <i>yaw</i>
Kode	UC-002
Kondisi awal	Pemain telah memasuki menu utama
Proses pengujian	Pengguna mngerakan kepala kearah kiri kekanan
Hasil yang diharapkan	Karakter melakukan pergerakan <i>yaw</i>
Hasil yang diperoleh	Karakter berhasil bergerak secara <i>yaw</i>
Kesimpulan	Berhasil

Gambar 6.2 Gambar implementasi *yaw*l

2. Pengujian Kepala secara *Pitch*

Tabel 6.5 Tabel Ujicoba UC-003

Nama Kasus Penggunaan	Mengerakan kepala secara <i>pitch</i>
Kode	UC-003
Kondisi awal	Pemain telah memasuki menu utama
Proses pengujian	Pengguna mnngerakan kepala kearah vertical
Hasil yang diharapkan	Karakter melakukan pergerakan <i>pitch</i>
Hasil yang diperoleh	Karakter berhasil bergerak secara <i>yaw</i>
Kesimpulan	Berhasil



Gambar 6.3 Gambar Implementasi *roll*

3. Pengujian Kepala secara *Roll*

Tabel 6.6 Tabel Ujicoba UC-004

Nama Kasus Penggunaan	Mengerakan kepala secara <i>roll</i>
Kode	UC-004
Kondisi awal	Pemain telah memasuki menu utama
Proses pengujian	Pengguna mnggerakan kepala dengan memiringkan kearah kiri dan kekanan
Hasil yang diharapkan	Karakter melakukan pergerakan <i>roll</i>
Hasil yang diperoleh	Karakter berhasil bergerak secara <i>roll</i>
Kesimpulan	Berhasil



Gambar 6.4 Gambar Implementasi *roll*

5.4.3 Skenario Pengujian Mendeteksi Emosi

1. Pengujian mendeteksi emosi *Happy*

Tabel 6.7 Tabel Ujicoba UC-005

Nama Kasus Penggunaan	Mendeteksi emosi <i>Happy</i>
Kode	UC-005
Kondisi awal	Pemain telah memasuki menu utama
Proses pengujian	Pengguna menampilkan emosi senang
Hasil yang diharapkan	Sistem menampilkan emosi <i>Happy</i>
Hasil yang diperoleh	Sistem berhasil menampilkan emosi <i>Happy</i>
Kesimpulan	Berhasil



Gambar 6.5 Gambar implementasi happy

2. Pengujian mendeteksi emosi *Sad*

Tabel 6.8 Tabel Ujicoba UC-006

Nama Kasus Penggunaan	Mendeteksi emosi <i>Sad</i>
Kode	UC-006
Kondisi awal	Pemain telah memasuki menu utama
Proses pengujian	Pengguna menampilkan emosi <i>sad</i>
Hasil yang diharapkan	Sistem menampilkan emosi <i>Sad</i>
Hasil yang diperoleh	Sistem berhasil menampilkan emosi <i>Sad</i>
Kesimpulan	Berhasil



Gambar 6.6 Gambar implementasi Sad

3. Pengujian mendeteksi emosi *Angry*

Tabel 6.9 Tabel Ujicoba UC-007

Nama Kasus Penggunaan	Mendeteksi emosi <i>Angry</i>
Kode	UC-007
Kondisi awal	Pemain telah memasuki menu utama
Proses pengujian	Pengguna menampilkan emosi <i>angry</i>
Hasil yang diharapkan	Sistem menampilkan emosi <i>Angry</i>
Hasil yang diperoleh	Sistem berhasil menampilkan emosi <i>Angry</i>
Kesimpulan	Berhasil

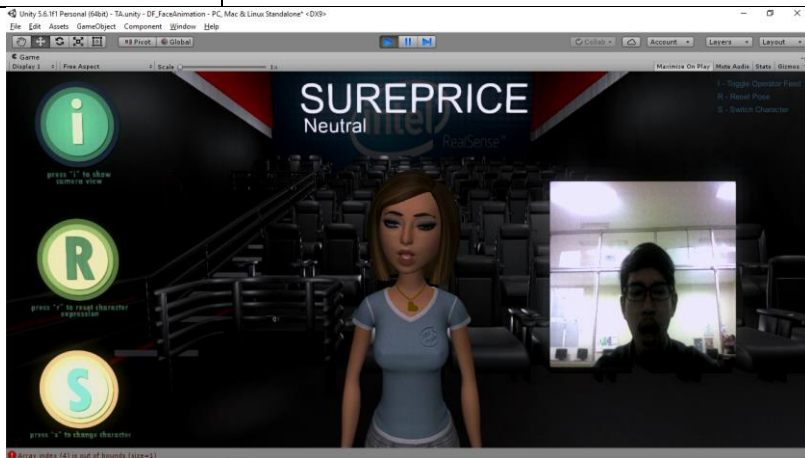


Gambar 6.7 Gambar implementasi *Angry*

4. Pengujian mendeteksi emosi *Sureprice*

Tabel 6.10 Tabel Ujicoba UC-008

Nama Kasus Penggunaan	Mendeteksi emosi <i>Sureprice</i>
Kode	UC-008
Kondisi awal	Pemain telah memasuki menu utama
Proses pengujian	Pengguna menampilkan emosi <i>Sureprice</i>
Hasil diharapkan yang	Sistem menampilkan emosi <i>Sureprice</i>
Hasil diperoleh yang	Sistem berhasil menampilkan emosi <i>Sureprice</i>
Kesimpulan	Berhasil



Gambar 6.8 Gambar implementasi Sureprice

5. Pengujian mendeteksi emosi *Fear*

Tabel 6.11 Tabel Ujicoba UC-009

Nama Kasus Penggunaan	Mendeteksi emosi <i>Fear</i>
Kode	UC-009
Kondisi awal	Pemain telah memasuki menu utama
Proses pengujian	Pengguna menampilkan emosi <i>Fear</i>
Hasil yang diharapkan	Sistem menampilkan emosi <i>Fear</i>
Hasil yang diperoleh	Sistem berhasil menampilkan emosi <i>Fear</i>
Kesimpulan	Berhasil



Gambar 6.9 Gambar implementasi Fear

6. Pengujian mendeteksi emosi *Disgust***Tabel 6.12 Tabel Ujicoba UC-010**

Nama Kasus Penggunaan	Mendeteksi emosi <i>Disgust</i>
Kode	UC-010
Kondisi awal	Pemain telah memasuki menu utama
Proses pengujian	Pengguna menampilkan emosi <i>Disgust</i>
Hasil yang diharapkan	Sistem menampilkan emosi <i>Disgust</i>
Hasil diperoleh	Sistem berhasil menampilkan emosi <i>Disgust</i>
Kesimpulan	Berhasil

**Gambar 6.10 Gambar implementasi Disgust**

5.4.4 Seknario Pengujian Mengganti Karater

Tabel 6.13 Ujicoba UC-011

Nama Kasus Penggunaan	Menggerakan wajah karakter
Kode	UC-011
Kondisi awal	Pemain telah memasuki menu utama
Proses pengujian	Pengguna menekan tombol i
Hasil yang diharapkan	Karakter berganti sesuai dengan urutan array karakter
Hasil yang diperoleh	Karakter berhasil diganti
Kesimpulan	Berhasil



Gambar 6.11 Hasil ujicoba sebelum mengganti karakter



Gambar 6.12 Hasil ujicoba setelah diganti karakter

5.4.5 Hasil Pengujian Fungsional

Subbab ini berisi tentang hasil pengujian fungsionalitas yang sudah dilakukan berdasarkan usecase UC-001 sampai dengan UC-004. Empat pengujian yang telah dilakukan menunjukkan bahwa program telah berjalan dengan baik. Rekapitulasi ditunjukkan pada Tabel 5.14:

Tabel 6.14 Hasil rekapitulasi

Nomor Pengujian	Hasil Pengujian
UC-001	Berhasil
UC-002	Berhasil
UC-003	Berhasil
UC-004	Berhasil

5.5 Pengujian Pengguna

Pengujian pada perangkat lunak yang dibangun tidak hanya digunakan pada fungsionalitas sendiri, tetapi ada juga pada pengguna untuk percobaan secara langsung. Pengujian ini berfungsi sebagai pengujian subjektif yang bertujuan untuk mengetahui tingkat keberhasilan aplikasi dari sudut pandang pengguna. Hal ini didapatkan dengan kuesioner yang diberikan kepada penguji.

Kuesioner mencakup semua penilaian yang nanti diberikan kepada penguji. Sistem penilaiannya berdasarkan ya dan tidak serta penilaian dari 1 sampai 6. Nilai dimulai dari yang paling rendah hingga yang paling tinggi. Tabel penilaian dapat dilihat pada Tabel 5.15:

Tabel 6.15 Tabel nilai pengujian

Nilai	Keterangan
1	Sangat tidak setuju
2	Tidak setuju
3	Kurang setuju
4	Cukup setuju
5	Setuju
6	Setuju

5.5.1 Hasil Penilaian

Hasil penilaian yang telah dilakukan melalui kuesioner yang telah dibagikan dapat dilihat pada Tabel 5.16:

Tabel 6.16 Hasil Tabel quisioner

No	Task	Penilaian					Rata
		1	2	3	4	5	Rata
1	Apakah aplikasi mudah digunakan?	6	5	6	6	6	5.8
2	Apakah tampilan aplikasi sudah baik?	5	5	5	6	5	5.2
3	Apakah aplikasi sudah nyaman digunakan?	5	5	5	5	5	5
4	Aplikasi dapat berjalan lancar tanpa adanya lag dan atau crash	5	5	4	5	4	4.6
5	Saya merasa terbantu dengan adanya petunjuk yang disediakan	4	5	5	5	6	5
6	Saya dapat menggunakan kamera, walaupun untuk pertama kalinya	5	5	5	6	6	5.4
7	Semua karakter bergerak sesuai dengan gerakan pengguna	4	4	5	5	5	4.6
8	Emosi dapat ditampilkan sesuai dengan yang diinginkan	4	3	5	5	4	4.2
Nilai total							4.475

5.5.2 Tanggapan pengguna

Pada kuesioner ini terdapat tanggapan pengguna berupa kritik dan saran untuk aplikasi virtual piano, pada subbab ini diunjukkan tanggapan responden setelah menggunakan aplikasi. Tabel tanggapan responden dapat dilihat pada Tabel 5.17:

Tabel 6.17 Tabel Tanggapan

No	Tanggapan
1	Tingkat intensitas pembeda antara satu ekspresi dengan ekspresi lainnya lebih dibedakan agar dapat mendeteksi lebih cepat lagi, <i>overall</i> bagus
2	Salah satu karakter masih <i>error</i> pergerakan kepala, tampilan bagus
3	Diperbanyak avatar

4	Aplikasi sudah berjalan lancar dan secara garis besar fungsi utama sudah jalan. Akan tetapi akan lebih baik lagi jika nantinya parameter penentu emosi lebih mendetail sehingga kamera dapat dengan cepat menangkap ekspresi dan menampilkan dengan lebih cepat
5	Perlu adanya pengecekan pada emosi karakter

5.5.3 Evaluasi pengujian

Berdasarkan hasil pengujian fungsionalitas, dan kuesioner yang telah dilakukan sebelumnya didapatkan bahwa:

4. Aplikasi berhasil mendeteksi wajah pengguna, dibuktikan dengan pada gambar saat menampilkan landmark. Hal ini dibuktikan pada pengujian UC-001.
5. Aplikasi berhasil mendeteksi emosi dari pengguna dibuktikan dengan pada gambar saat menampilkan emosi pada aplikasi. Hal ini dibuktikan pada pengujian UC-001 sampai dengan UC-0010.
6. Berdasarkan kuesioner, tampilan aplikasi sudah menarik dan nyaman digunakan dibuktikan pada Tabel 5.9.
7. Masih terdapat *error* pada aplikasi terutama pada kurang tepatnya pendeteksi emosi serta pergerakan karakter yang masih ada yang salah.

(Halaman ini sengaja dikosongkan)

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari tujuan pembuatan perangkat lunak dan hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, terdapat pula saran yang ditujukan untuk pengembangan perangkat lunak lebih lanjut.

6.1 Kesimpulan

Dalam proses pengerjaan tugas akhir mulai dari tahap analisis, desain, implementasi, hingga pengujian didapatkan kesimpulan sebagai berikut:

1. Aplikasi *faceshift* yang dibangun menggunakan teknologi *Intel Realsense SDK* dapat mendeteksi emosi pengguna walaupun dengan akurasi yang kurang tepat dikarenakan, kamera *Intel Realsense* sering memiliki nilai *intensity* yang sama pada dua atau lebih emosi sehingga salah mengartikan emosi dari pengguna.
2. Aplikasi ini menyediakan tampilan yang menarik dan mudah dimengerti.

6.2 Saran

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang, berdasarkan pada hasil perancangan, implementasi dan uji coba yang telah dilakukan.

1. Memperbaiki *Intel Realsense SDK* menjadi lebih baik agar memperbaiki kesalahan, error, serta meningkatkan sensitivitas, pengaturan cahaya yang dibutuhkan dan kualitas kamera.
2. Mencari lokasi yang memiliki penarangan yang baik
3. Dibutuhkan penelitian yang spesifik mengenai penggunaan SDK yang berbeda agar dapat mengembangkan aplikasi lebih baik lagi.

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

- [1] "Computer Animation," 2015. [Online]. Available: https://en.wikipedia.org/wiki/Computer_animation. [Accessed 3 12 2016].

- [2] Vinícius Silva , Filomena Soares, João S. Esteves , Joana Figueiredo, "Real-time Emotions Recognition System", Industrial Electronics Department, School of Engineering, 2016.

- [3] "Intel RealSense," 12 10 2016. [Online]. Available: https://en.wikipedia.org/wiki/Intel_RealSense. [Accessed 3 12 2016].

- [4] Unity, "CREATE THE GAMES YOU LOVE WITH UNITY," Unity Technologies, 2014. [Online]. Available: <http://unity3d.com/unity>. [Accessed 22 12 2015].

- [5] "Blender (software)," Wikipedia, 19 12 2016. [Online]. Available: [https://en.wikipedia.org/wiki/Blender_\(software\)](https://en.wikipedia.org/wiki/Blender_(software)). [Accessed 16 Agustus 2016].

- [6] I. S. Mackenzie, "Human-Computer Interaction: An Empirical Research Perspective," Morgan Kaufmann, 2013.

(Halaman ini sengaja dikosongkan)

LAMPIRAN



Kuisisioner Tugas Akhir "Rancang Bangun Aplikasi Faceshift Menggunakan Motion Capture pada Wajah dengan Teknologi Intel Realsense"

Nama : *Rahmat Rizal*
 NRP : *5119100706*
 Pekerjaan : *Mahasiswa*

A. KARAKTERISTIK RESPONDEN

No	Pertanyaan	YA	Tidak
1	Apakah anda mengetahui tentang intel realsense kamera?	✓	
2	Apakah anda Pernah menggunakan tentang intel realsense kamera?		✓

Silahkan centang pada kolom yang sesuai berikut ini :

No	Pertanyaan	Nilai					
		1	2	3	4	5	6
1	Apakah aplikasi mudah digunakan?						✓
2	Apakah tampilan aplikasi sudah baik?					✓	
3	Apakah aplikasi sudah nyaman digunakan?					✓	
4	Aplikasi dapat berjalan lancar tanpa adanya lag dan atau crash				✓		
5	Saya merasa terbantu dengan adanya petunjuk yang disediakan						✓
6	Saya dapat menggunakan kamera, walaupun untuk pertama kalinya						✓
7	Semua karakter bergerak sesuai dengan gerakan pengguna					✓	
8	Emosi dapat ditampilkan sesuai dengan yang diinginkan				✓		

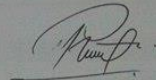
Keterangan :

1. Sangat Tidak Setuju 2. Tidak Setuju 3. Kurang Setuju 4. Cukup Setuju 5. Setuju 6. Sangat Setuju

Kritik dan Saran untuk pengembangan selanjutnya:

Tingkat interrelasi Pembida antara satu ekspansi dengan ekspansi yang lebih sebelumnya agar dapat menghasilkan lebih awal lagi order all udah bagus.

Surabaya, 09 Juni 2017



(Rahmat Rizal)

Kuisiner Tugas Akhir "Rancang Bangun Aplikasi Faceshift Menggunakan Motion Capture pada Wajah dengan Teknologi Intel Realsense"

Nama : *Tika I. May*
 NRP : *511310069*
 Pekerjaan : *Mahasiswa*

A. KARAKTERISTIK RESPONDEN

No	Pertanyaan	YA	Tidak
1	Apakah anda mengetahui tentang intel realsense kamera?		✓
2	Apakah anda Pernah menggunakan tentang intel realsense kamera?		✓

Silahkan centang pada kolom yang sesuai berikut ini :

No.	Pertanyaan	Nilai					
		1	2	3	4	5	6
1	Apakah aplikasi mudah digunakan?						✓
2	Apakah tampilan aplikasi sudah baik?						✓
3	Apakah aplikasi sudah nyaman digunakan?					✓	
4	Aplikasi dapat berjalan lancar tanpa adanya lag dan atau crash					✓	
5	Saya merasa terbantu dengan adanya petunjuk yang disediakan					✓	
6	Saya dapat menggunakan kamera, walaupun untuk pertama kalinya						✓
7	Semua karakter bergerak sesuai dengan gerakan pengguna					✓	
8	Emosi dapat ditampilkan sesuai dengan yang diinginkan					✓	

Keterangan :

1. Sangat Tidak Setuju 2. Tidak Setuju 3. Kurang Setuju 4. Cukup Setuju 5. Setuju 6. Sangat Setuju

Kritik dan Saran untuk pengembangan selanjutnya:

Kepala berputar-putar pada setiap satu karakter. Tampilan bagus

Surabaya, 9 Juni 2017

Ahr

(Tika I. May)



Kuisisioner Tugas Akhir "Rancang Bangun Aplikasi Faceshift Menggunakan Motion Capture pada Wajah dengan Teknologi Intel Realsense"

Nama : Naupal B. Fauzan

NRP : 5113100175

Pekerjaan : Mahasiswa

A. KARAKTERISTIK RESPONDEN

No	Pertanyaan	YA	Tidak
1	Apakah anda mengetahui tentang intel realsense kamera?	✓	
2	Apakah anda Pernah menggunakan tentang intel realsense kamera?		✓

Silahkan centang pada kolom yang sesuai berikut ini :

No.	Pertanyaan	Nilai					
		1	2	3	4	5	6
1	Apakah aplikasi mudah digunakan?						✓
2	Apakah tampilan aplikasi sudah baik?					✓	
3	Apakah aplikasi sudah nyaman digunakan?					✓	
4	Aplikasi dapat berjalan lancar tanpa adanya lag dan atau crash					✓	
5	Saya merasa terbantu dengan adanya petunjuk yang disediakan				✓		
6	Saya dapat menggunakan kamera, walaupun untuk pertama kalinya					✓	
7	Semua karakter bergerak sesuai dengan gerakan pengguna				✓		
8	Emosi dapat ditampilkan sesuai dengan yang diinginkan				✓		

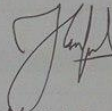
Keterangan :

1. Sangat Tidak Setuju 2. Tidak Setuju 3. Kurang Setuju 4. Cukup Setuju 5. Setuju 6. Sangat Setuju

Kritik dan Saran untuk pengembangan selanjutnya:

Perlu disediakan pengendalian pada perubahan emosi karakter

Surabaya, 5 Juni 2017



(Naypol Bayu Fauzan)



Kuisisioner Tugas Akhir "Rancang Bangun Aplikasi Faceshift Menggunakan Motion Capture pada Wajah dengan Teknologi Intel Realsense"

Nama : Guruh Anya Senna

NRP : 5113100 010

Pekerjaan : Mahasiswara

A. KARAKTERISTIK RESPONDEN

No	Pertanyaan	YA	Tidak
1	Apakah anda mengetahui tentang intel realsense kamera?	✓	
2	Apakah anda Pernah menggunakan tentang intel realsense kamera?	✓	

Silahkan centang pada kolom yang sesuai berikut ini :

No.	Pertanyaan	Nilai					
		1	2	3	4	5	6
1	Apakah aplikasi mudah digunakan?					✓	
2	Apakah tampilan aplikasi sudah baik?					✓	
3	Apakah aplikasi sudah nyaman digunakan?					✓	
4	Aplikasi dapat berjalan lancar tanpa adanya lag dan atau crash					✓	
5	Saya merasa terbantu dengan adanya petunjuk yang disediakan					✓	
6	Saya dapat menggunakan kamera, walaupun untuk pertama kalinya					✓	
7	Semua karakter bergerak sesuai dengan gerakan pengguna				✓		
8	Emosi dapat ditampilkan sesuai dengan yang diinginkan			✓			

Keterangan :

1. Sangat Tidak Setuju 2. Tidak Setuju 3. Kurang Setuju 4. Cukup Setuju 5. Setuju 6. Sangat Setuju

Kritik dan Saran untuk pengembangan selanjutnya:

Aplikasi sudah berjalan lancar dan secara garis besar sudah fungsi utamanya sudah berjalan. Akan lebih baik lagi jika nantinya parameter-parameter penting emosi lebih mendalam sehingga kamera dengan cepat dapat menangkap ekspresi dan aplikasi menampilkan jenis emosi yang sesuai dengan cepat pula.

Surabaya, 4 Juni 2017

1. Guruh Arya Senna

Kuisisioner Tugas Akhir "Rancang Bangun Aplikasi Faceshift Menggunakan Motion Capture pada Wajah dengan Teknologi Intel Realsense"

Nama : Fajar Ade Putra
 NRP : 5113100092
 Pekerjaan : Mahasiswa

A. KARAKTERISTIK RESPONDEN

No	Pertanyaan	YA	Tidak
1	Apakah anda mengetahui tentang intel realsense kamera?	✓	
2	Apakah anda Pernah menggunakan tentang intel realsense kamera?		✓

Silahkan centang pada kolom yang sesuai berikut ini :

No.	Pertanyaan	Nilai					
		1	2	3	4	5	6
1	Apakah aplikasi mudah digunakan?						✓
2	Apakah tampilan aplikasi sudah baik?					✓	
3	Apakah aplikasi sudah nyaman digunakan?					✓	
4	Aplikasi dapat berjalan lancar tanpa adanya lag dan atau crash				✓		
5	Saya merasa terbantu dengan adanya petunjuk yang disediakan					✓	
6	Saya dapat menggunakan kamera, walaupun untuk pertama kalinya					✓	
7	Semua karakter bergerak sesuai dengan gerakan pengguna					✓	
8	Emosi dapat ditampilkan sesuai dengan yang diinginkan					✓	

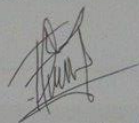
Keterangan :

1. Sangat Tidak Setuju 2. Tidak Setuju 3. Kurang Setuju 4. Cukup Setuju 5. Setuju 6. Sangat Setuju

Kritik dan Saran untuk pengembangan selanjutnya:

lebih banyak lagi contohnya

Surabaya, 9 Juni 2017



(Fajar Ade Putra ,

BIODATA PENULIS



Penulis lahir di Palembang, 30 Maret 1996 merupakan anak kedua dari 3 bersaudara. Dalam menjalani pendidikan semasa hidup, penulis menempuh pendidikan di SD PUSRI Kota Palembang, SMP PUSRI Kota Palembang, SMAN 5 Palembang, dan S1 Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember (ITS) pada rumpun Interaksi Grafika dan Seni (IGS).

Di jurusan Teknik Informatika, penulis mengambil bidang minat IGS dan memiliki ketertarikan dengan bidang seni, desain web dan permodelan 3D. Penulis aktif sebagai staf Departemen PSDM HMTTC ITS periode 2014-2015, staf Departemen OSR BEM FTIF ITS periode 2014-2015, Selain itu penulis pernah menjadi wakil eksternal BEM FTIf ITS 2015-2016 dan pernah kerja praktek di perusahaan Metrodata Jakarta. Penulis dapat dihubungi melalui email ishardanardan@outlook.com

